

N O T I C E

THIS DOCUMENT HAS BEEN REPRODUCED FROM
MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT
CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED
IN THE INTEREST OF MAKING AVAILABLE AS MUCH
INFORMATION AS POSSIBLE

NF
NASA Technical Memorandum 73294

Radar Altimeter Waveform Modeled Parameter Recovery

(NASA-TM-73294) RADAR ALTIMETER WAVEFORM
MODELED PARAMETER RECOVERY (NASA) 59 p
HC A04/MF A01 CSCL 171

N81-30325

Unclass

G3/32 27228

G. S. Hayne

August 1981

NASA

National Aeronautics and
Space Administration

Wallops Flight Center
Wallops Island, Virginia 23337
AC 804 824-3411

NASA Technical Memorandum 73294

Radar Altimeter Waveform Modeled Parameter Recovery

G. S. Hayne

**NASA Wallops Flight Center
Wallops Island, Virginia 23337**



**National Aeronautics and
Space Administration**

**Wallops Flight Center
Wallops Island, Virginia 23337
AC 804 824-3411**

RADAR ALTIMETER WAVEFORM MODELED PARAMETER RECOVERY

George S. Hayne
NASA Wallops Flight Center
Wallops Island, VA 23337

INTRODUCTION

Recent satellite-borne radar altimeters such as GEOS-3 and SEASAT-1 have included waveform sampling gates providing point samples of the transmitted radar pulse after it has been scattered from the ocean's surface. There were 16 such samplers in GEOS-3 and 63 on SEASAT-1, and Table I lists the time locations of the 63 SEASAT-1 samplers relative to the middle one of the set. The waveform sampler data can be averaged over some specified number of individual radar returns, and the resulting set of average values can be best-fitted in some sense by varying the parameters in a model or theoretical mean return waveform. This report describes the waveform model used, and the details of fitting this model to the waveform data to obtain estimates of the parameters characterizing the modeled waveform.

The particular model radar return waveform used at Wallops Flight Center for SEASAT data analysis is characterized by six parameters: 1) amplitude; 2) time-origin, or track-point; 3) ocean surface rms roughness; 4) noise baseline; 5) ocean surface skewness; and 6) attitude angle, or off-nadir angle. Figure 1 sketches the model waveform and indicates qualitatively how the model shape changes as these parameters vary. These will be discussed later, but we can briefly state here the importance of these estimated parameters.

The time-origin parameter is the location of the actual mean return waveform relative to the altimeter's altitude-tracker-positioned sampling gate set, and the time-origin is thus directly interpretable as an altitude correction to be applied to the real-time altitude output. The ocean surface rms roughness provides a revised estimate of the significant waveheight (SWH). The ocean surface skewness parameter provides additional information about the surface elevation probability density function and possibly also about the ocean wave spectrum. The amplitude parameter may be used to revise the altimeter-estimated surface backscattering cross-section σ^0 , and the attitude angle also leads to a correction to σ^0 . The noise baseline parameter is of relatively little direct interest but must be included as one of the fitted waveform parameters because the sampling gates measure radar signal plus noise.

TABLE 1. TIME LOCATION AND INDEXING FOR THE 63 SEASAT WAVEFORM SAMPLERS

<u>Index</u>	<u>Time, ns</u>	<u>Previous SEASAT #</u>	<u>#, This Report</u>	<u>Index</u>	<u>Time, ns</u>	<u>Previous SEASAT #</u>	<u>#, This Report</u>
1	-92.1875	-30	1	35	4.6875	+ 2	32
2	-89.0625	-29	2	36	7.8125	+ 3	33
3	-85.9375	-28	3	37	10.9375	+ 4	34
4	-82.8125	-27	4	38	14.0625	+ 5	35
5	-79.6875	-26	5	39	17.1875	+ 6	36
6	-76.5625	-25	6	40	20.3125	+ 7	37
7	-73.4375	-24	7	41	23.4375	+ 8	38
8	-70.3125	-23	8	42	26.5625	+ 9	39
9	-67.1875	-22	9	43	29.6875	+10	40
10	-64.0625	-21	10	44	32.8125	+11	41
11	-60.9375	-20	11	45	35.9375	+12	42
12	-57.8125	-19	12	46	39.0625	+13	43
13	-54.6875	-18	13	47	42.1875	+14	44
14	-51.5625	-17	14	48	45.3125	+15	45
15	-48.4375	-16	15	49	48.4375	+16	46
16	-45.3125	-15	16	50	51.5625	+17	47
17	-42.1875	-14	17	51	54.6875	+18	48
18	-39.0625	-13	18	52	57.8125	+19	49
19	-35.9375	-12	19	53	60.9375	+20	50
20	-32.8125	-11	20	54	64.0625	+21	51
21	-29.6875	-10	21	55	67.1875	+22	52
22	-26.5625	- 9	22	56	70.3125	+23	53
23	-23.4375	- 8	23	57	73.4375	+24	54
24	-20.3125	- 7	24	58	76.5625	+25	55
25	-17.1875	- 6	25	59	79.6875	+26	56
26	-14.0625	- 5	26	60	82.8125	+27	57
27	-10.9375	- 4	27	61	85.9375	+28	58
28	- 7.8125	- 3	28	62	89.0625	+29	59
29	- 4.6875	- 2	29	63	92.1875	+30	60
30	- 3.1250	- 1½	29½				
31	- 1.5625	- 1	30				
32	0.0000	0	30½				
33	1.5625	+ 1	31				
34	3.1250	+ 1½	31½				

The SEASAT fitted waveform is characterized by the following 6 parameters:

Amplitude,

Time origin,

Ocean surface rms elevation σ_s ,

Baseline,

Ocean surface skewness, and

Off-nadir angle ξ .

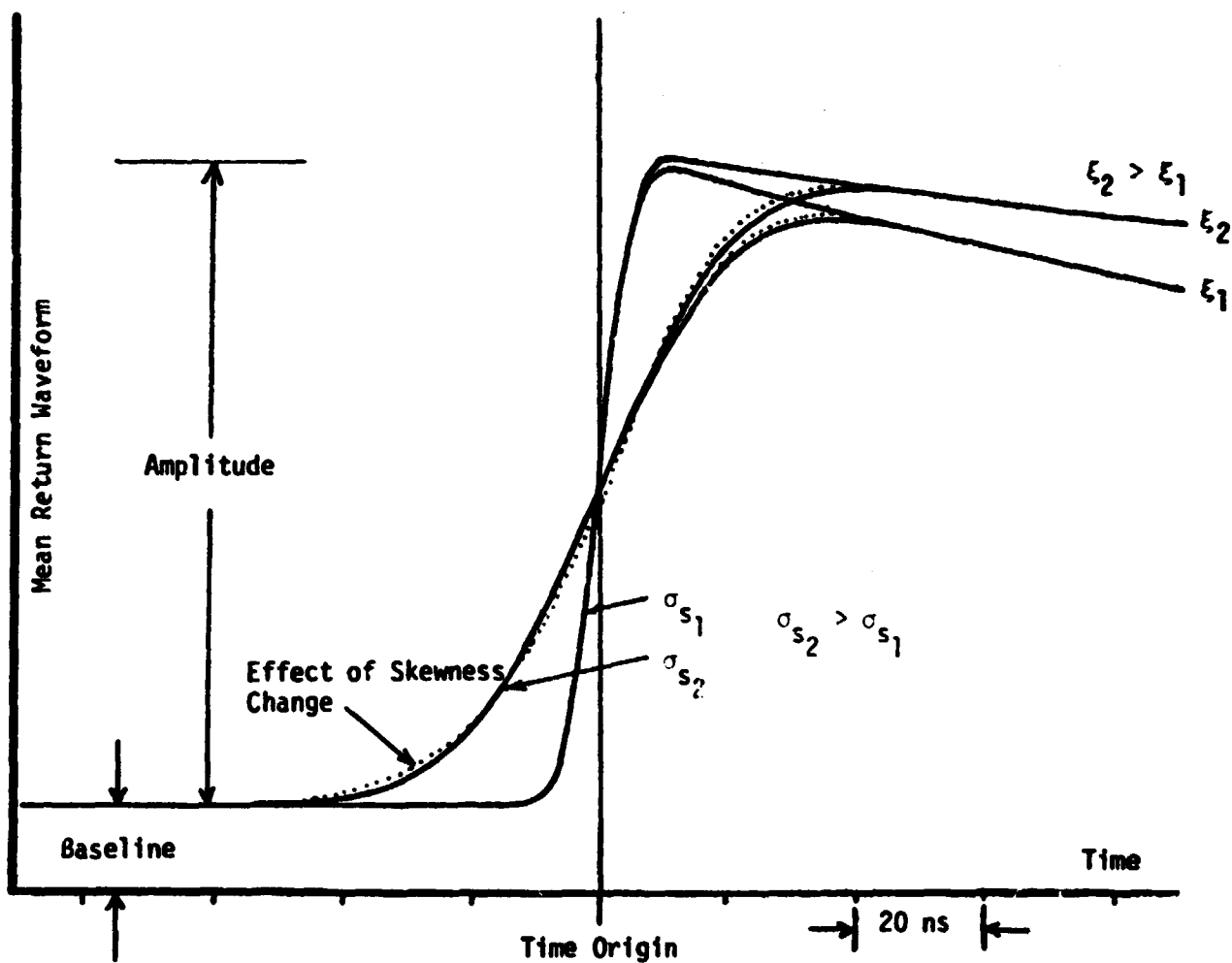


Figure 1. SEASAT model waveform.

This report is intended to be relatively complete and self-contained in describing the waveform model and the parameter-estimating procedure. The work depends of course on a number of other reports and papers which will be referred to in the text, but all necessary mathematics should be contained in the report or in the program source listings in the Appendix sections. The actual computer implementation is relatively simple but it is not so simple to describe succinctly the number of judgments and decisions built into the work. I have tried to refer to some of these considerations in this report and this is my partial excuse for a more wordy account than that of a typical technical journal article.

There are several separate categories which would be included in a complete account of our SEASAT data analysis, including the following topics:

- 1) details of waveform data input to the programs, data calibration and correction, data editing or screening, the waveform averaging period, and so forth,
- 2) the general procedure used in fitting a several-parameter function to the corrected waveform sampler averages,
- 3) the details of and justification for the model waveform used as the several-parameter function in the fitting procedure,
- 4) verification of the model and of the fitting programs, and
- 5) presentation of specific results and ground truth comparisons for the results of the data analysis.

This report is not a complete SEASAT data study and will not present any of the information in 5) above. Item 1) will be ignored except to note that most of our SEASAT work typically used 10,000 pulse, ten second waveform data averages. Item 4) on verification is only briefly discussed later. This report will concentrate on items 2) and 3) in the following two sections, describing first the general iterative non-linear least-squares fitting procedure and then the model waveform used. At the heart of any waveform data fitting procedure is the theoretical model which is being fitted, and this report places its major emphasis on that topic.

As another way of establishing the intended purpose of this report, Figure 2 shows the various procedures which should be included within the general category of waveform processing in planning the data processing for any future satellite-borne radar altimeter. Figure 2 is a rough sketch based on some current Wallops Flight Center studies for the proposed TOPEX radar altimeter program, but the figure is general and not restricted to any one specific program. Outputs from the general waveform processor of Figure 2 include: 1) new quantities not otherwise available (such as ocean surface skewness); 2) quantities which are replacements for, and are more precise than, radar altimeter real-time estimates (such as SWH); and 3) quantities which are corrections to altimeter real-time estimates (such as altitudes). Within the general waveform processor of Figure 2 is a box labeled Waveform Modeled Parameter Recovery, and this present report is concerned only with this one particular box in the general Figure 2.

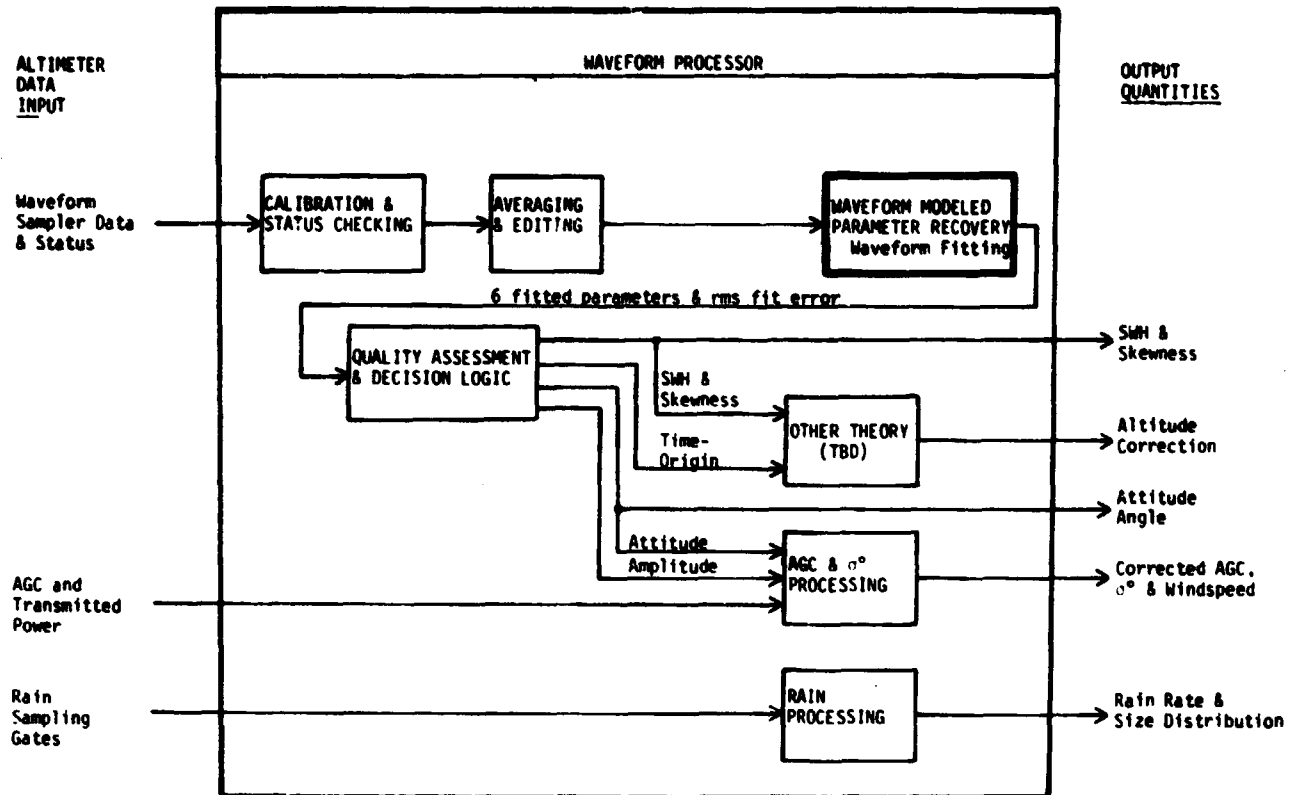


Figure 2. Procedures within overall waveform processor.

There are a couple of points of general philosophy to mention here. First, we have strongly preferred fitting the data from all of the SEASAT waveform samplers in the plateau region of the return pulse. Otherwise one would need to decide what samplers to ignore, and this criterion would probably have to be a function of significant waveheight. Second, the model waveform is the convolution of several terms as described later, and the parameters of interest could be obtained by some type of deconvolution process as done by Lipa and Barrick (ref. 1). A direct (numerical) deconvolution in this type of problem, a problem characterized by an inherently noisy set of discrete observations over a finite time span, has the general reputation of significant numerical difficulties and instabilities. Accordingly, at the onset of this work some years ago, we chose to use what could be characterized as "iterative convolution," in which an assumed functional form is fitted through a non-linear least-squares procedure, and this is the method described in the remainder of this report.

Finally, a brief paragraph on the history of the SEASAT analysis and on the relationship of this report to other work may be useful. There has been an active effort in waveform analysis at Wallops Flight Center since before the GEOS-3 satellite was launched. The waveform analysis used here for GEOS-3 was described briefly in an earlier report of mine (ref. 2), and this algorithm is among several which were compared in a paper by Fedor et al. (ref. 3). Our earlier GEOS-3 work and the present SEASAT-1 work are based on the radar-ocean surface interaction summarized by Brown (ref. 4). In a recent paper (ref. 5), I developed a several-term expansion describing the mean return waveform from an idealized SEASAT-like radar altimeter; the paper's results do not apply directly to the actual SEASAT-1 radar altimeter because the actual SEASAT-1 radar pulse (sampled by a calibration mode) is not the simple form assumed in Reference 5. Consequences of the actual SEASAT-1 point target response and of certain calibration problems are described in a report (ref. 6), and preliminary results of our waveform processing of SEASAT-1 data are in References 7 and 8. Details of the actual SEASAT-1 altimeter design and altimeter hardware are available in References 9 and 10, and some of the standard ground-based data processing for SEASAT-1 data is described by Hancock et al. (ref. 11). A later section of this report discusses numerical convolutions performed through use of the Fast Fourier Transform, and Brigham (ref. 12) is one of many sources of information on this subject. We have already mentioned the somewhat different approach to waveform processing by Lipa and Barrick (ref. 1). The principal goal in our work has been the estimation of sea surface skewness from satellite altimeter data, and the work of Huang and colleagues (refs. 13 and 14) provides a relationship between ocean surface spectrum details and skewness and significant waveheight.

NON-LINEAR LEAST-SQUARES FUNCTION-FITTING ROUTINE

Introductory Discussion

The problem of fitting a function to input experimental data has been divided into two pieces; this section of the report will describe the fitting procedure itself with little concern for what function is being fitted, while a later section of the report will describe the specific function (the model waveform) to be fitted in the radar altimeter waveform problem.

A least-squares fitting procedure is desired but there is a limited number of types of functions for which the normal equations are directly solvable in an ordinary least-squares procedure. The model waveform function described later in this report produces an intractable set of normal equations, and in this situation the approach is to: a) make a

first guess at the values of the parameters in the fitting function, b) do a first-order Taylor series expansion in parameter space around the first guess values, c) do a least-squares solution for the corrections to the first-guess parameter values, and d) use the revised parameter values as a new first guess, looping back to step b) and repeating until some exit criterion is satisfied. This was used earlier in the GEOS-3 data analysis (ref. 2) and the general approach has been described in a number of different sources such as Reference 15. (References 16 and 17 provide some additional discussion of some interest to this general problem.) The following subsection will present this method in more specific detail and a subsequent subsection then discusses the particular subroutine version implemented here.

General Derivation

Suppose that there is a set of experimental data Y_i , $i = 1$ to N , with each Y_i from a different value of an independent variable t_i . Suppose further that a model function R is to be fitted to these data, where R is described by a set of constants K and a set of m parameters Θ to be fitted,

$$\Theta = \{\theta_1, \theta_2, \dots, \theta_m\} . \quad (1)$$

We will designate the model function value at t_i by $R(t_i; K, \Theta)$. The number of points N must be greater than the number of parameters being fitted, m . For SEASAT, R will be the model return waveform, the Y_i will be the $N = 63$ waveform sampler values, and number of parameters fitted will be $m = 6$, but the derivation in this section is intended to be general and not specific to SEASAT alone.

Let Θ_0 be the set of initial guesses for the m parameters,

$$\Theta_0 = \{\theta_{1_0}, \theta_{2_0}, \dots, \theta_{m_0}\} . \quad (2)$$

and expand R in a Taylor series about the point Θ_0 in the m -dimensional fit parameter space. For notation convenience, define

$$R_{i_0} = R(t_i; K, \Theta_0) . \quad (3)$$

and the Taylor series is given by

$$R(t_i; K, \Theta) = R_{i_0} + (\theta_1 - \theta_{1_0}) \frac{\partial R_{i_0}}{\partial \theta_1} + (\theta_2 - \theta_{2_0}) \frac{\partial R_{i_0}}{\partial \theta_2} + \dots + (\theta_m - \theta_{m_0}) \frac{\partial R_{i_0}}{\partial \theta_m} + \text{higher-order terms.} \quad (4)$$

Define D_{ij} and S_j by

$$D_{ij} = \frac{\partial R_{i_0}}{\partial \theta_j} = \left. \frac{\partial}{\partial \theta_j} R(t_i; K, \Theta) \right|_{\text{evaluated at } \Theta_0} \quad (5)$$

and

$$\delta_j = \theta_j - \theta_{j_0} \quad (6)$$

ignore all the higher-order terms, and the above Taylor series becomes

$$R(t_i; K, \Theta) = R_{i_0} + \delta_1 D_{i1} + \delta_2 D_{i2} + \dots + \delta_m D_{im} \quad (7)$$

This is a linear equation in $\delta_1, \delta_2, \dots, \delta_m$ and can be treated by a least-squares method to solve for the $\delta_1, \dots, \delta_m$.

Although most of our SEASAT analysis has been for uniformly weighted data we will include the possibility of nonuniform data weighting in the following. Define a set of weights w_i for the (Y_i, t_i) experimental data points, and then define Q as the weighted sum of squares of the residuals between the Y_i and the $R(t_i; K, \Theta)$,

$$Q = \sum_{i=1}^N w_i [Y_i - R(t_i; K, \Theta)]^2$$

$$= \sum w_i [Y_i - R_{i_0} - \delta_1 D_{i1} - \delta_2 D_{i2} - \dots - \delta_m D_{im}]^2 \quad (8)$$

A minimum in Q is found by taking the partial derivatives of Q with respect to $\delta_1, \delta_2, \dots, \delta_m$, and setting each of these to zero. This set of m partial derivatives is

$$\begin{aligned}
\frac{\partial Q}{\partial \delta_1} &= -2 \sum w_i D_{i1} (Y_i - R_{i0} - \delta_1 D_{i1} - \delta_2 D_{i2} - \dots - \delta_m D_{im}) = 0 \\
\frac{\partial Q}{\partial \delta_2} &= -2 \sum w_i D_{i2} (Y_i - R_{i0} - \delta_1 D_{i1} - \delta_2 D_{i2} - \dots - \delta_m D_{im}) = 0 \\
&\vdots \\
&\vdots \\
&\vdots \\
\frac{\partial Q}{\partial \delta_m} &= -2 \sum w_i D_{im} (Y_i - R_{i0} - \delta_1 D_{i1} - \delta_2 D_{i2} - \dots - \delta_m D_{im}) = 0
\end{aligned} \tag{9}$$

All the indicated sums are from $i = 1$ to N . These m equations can then be written as a matrix equation,

$$B = A \Delta \tag{10}$$

in which B and Δ are m by 1 column matrices and A is a symmetric m by m matrix. These matrices are

$$\Delta = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \vdots \\ \vdots \\ \delta_m \end{bmatrix}, \quad B = \begin{bmatrix} \sum w_i D_{i1} (Y_i - R_{i0}) \\ \sum w_i D_{i2} (Y_i - R_{i0}) \\ \vdots \\ \vdots \\ \vdots \\ \sum w_i D_{im} (Y_i - R_{i0}) \end{bmatrix} \tag{11}$$

and

$$A = \begin{bmatrix} \sum w_i D_{i1}^2 & \sum w_i D_{i1} D_{i2} & \dots & \sum w_i D_{i1} D_{im} \\ \sum w_i D_{i1} D_{i2} & \sum w_i D_{i2}^2 & \dots & \sum w_i D_{i2} D_{im} \\ \vdots & \vdots & \ddots & \vdots \\ \sum w_i D_{i1} D_{im} & \sum w_i D_{i2} D_{im} & \dots & \sum w_i D_{im}^2 \end{bmatrix} \tag{12}$$

Then if matrix A has an inverse A^{-1} , one solves for A^{-1} and the solution for Δ is

$$\Delta = A^{-1} B. \quad (13)$$

Then from the definition of the δ_j in equation (6), the new parameter estimates θ_j can be obtained from δ_j and θ_{j_0} by

$$\theta_j = \theta_{j_0} + \delta_j, \quad j = 1 \text{ to } m. \quad (14)$$

These new θ_j parameters are not necessarily equal to the "true" or final parameter set θ because of the higher order terms having been ignored in the Taylor series leading to equation (7). The next step is to use the θ_j of equation (14) as a revised guess, a revised θ_0 , and to repeat the entire process just described.

This iterative process is repeated for some preset maximum number of iterations or until some type of convergence criterion is satisfied. Additional error exit provisions must be made for cases in which the matrix A is nearly singular or in which Q, the weighted sum of residuals squared, increases over its value in the preceding iteration. The usual convergence criterion is that the fractional change in Q from one iteration to the next should be less than some preset limit. An additional iteration exit should be provided for the absolute value of Q falling below some lower limit even if the Q fractional change criterion has not been met. The particular lower limit specified is based on reasonable judgment in the particular application of the general fitting procedure, and also can be influenced by computer precision limits due to finite computer word lengths.

Additional Discussion of the Wallops Version of a Non-linear Least-Squares Subroutine

The subroutine SMHFIT in Appendix B carries out the general iterative least-squares procedure described above. These several paragraphs will discuss some of the features and considerations in this subroutine.

The function being fitted is provided by a subroutine FILLV which is described further in this report's section on the model waveform. The various partial derivatives with respect to the fitted parameters, the D_{ij} of equation (5), are supplied by a subroutine FILLD. Notice that by keeping FILLV and FILLD outside the matrix-fitting loop within SMHFIT, we have the ability to modify the model waveform function at will without disturbing the fitting procedure; also the FILLV function is evaluated for all N (usually 63) values at one point in SMHFIT so that the convolution-by-FFT can be used as described later. For our recent work, the derivatives D_{ij} in FILLD are computed by making a step-

change in each of the estimated fit parameters and recalculating the fit function in FILLV, and the array SIPRM in SWHFIT contains the step-sizes currently used.

SWHFIT has been a general research tool for a variety of waveform work and we have found it useful to be able to change the order and number of parameters being fitted without having to change the program source code; an array JORDR allows this as described further by comments in the Appendix B source listing.

It has sometimes been useful to limit the size of the corrections Δ allowed in equation (14) for the first several iterations to prevent the iterative process from jumping completely outside the region of convergence in the event of a particularly poor first guess for the parameters θ_0 . Subroutine SWHFIT allows 1/5 the correction for the first iteration, 2/5 for the next, and so forth until the fifth and subsequent iterations when the full correction is made.

Another feature of SWHFIT is its provision for parameter constraints as suggested to us by Dr. William Wells of Washington Analytical Services Center, Inc. (of EG&G, Inc.). These constraints are added to the diagonal elements in the matrix A, with each constraint being approximately the inverse of the appropriate fit parameter's input variance estimate in cases in which some a priori information is available on the variation expected in the parameters to be fitted.

A general simulation program, to be described only briefly later in this report, has been used as an "experimental" means of setting the constraints used for the SEASAT analysis. The ad hoc recipe for limiting Δ in the first several iterations was the result of earlier data analysis and simulation. The exit criterion, based on the fractional change in the sum of residuals squared, has the value 0.001 in the SEASAT work, and the maximum number of iterations allowed is 30. SWHFIT contains features which are specific to the FILLV function being a radar altimeter mean return waveform. One such feature is a test which bars the possibility of a negative risetime or a negative significant wave-height. Notice also in SWHFIT that when the antenna pointing angle is being fitted, there are limits on the range of values allowed. Also in the case that the sum of residuals squared increases after several iterations, the parameter values returned by SWHFIT will be those for which the sum of residuals squared had its minimum.

For the general iterative non-linear least-squares procedure described, there is no fundamental guarantee that the sum of residuals squared which is reached in the fit is the global minimum and not just a local minimum. We can only point out that for a number of cases tried in the simulation and verification procedure, both with and without noise added to the simulated waveform samples, the correct parameters were recovered in the fitting procedure. Furthermore, the results seemed insensitive to a range of first-guess values of the parameters.

MODEL WAVEFORM FOR RADAR ALTIMETER DATA

In this section I will briefly review the fundamental assumptions underlying the model waveform used for fitting to the radar altimeter waveform sampler data. The three separate terms to be convolved in the model are then separately discussed. One of these terms has, for SEASAT, been impossible to represent by a simple analytical expression with the consequence that at least one numerical convolution step is required in generating a model waveform for any particular set of waveform parameters. To reduce the computation time required, this numerical convolution step can be performed through Fast Fourier Transform (FFT) techniques, and this is further described in a later subsection of this section on the model waveform.

Convolutional Model for Rough Surface Radar Returns

In 1957 Moore and Williams (ref. 18) demonstrated that the average radar power return from a rough surface and for near-normal incidence could be expressed as a convolution of the "transmitted pulse shape" and a term which included the effects of antenna pattern and off-nadir pointing angle, surface properties, and distance. More recently, Brown (ref. 4) discussed this convolutional model and, for the assumptions common to satellite radar altimetry systems, produced a simplified closed-form expression for the average rough surface impulse response function.

My own work is based on the material in Brown's paper (ref. 4) as a starting point and I will assume that the average waveform can be viewed in an "elapsed-time" domain in which the actual hardware waveform sampling gates can be labelled in successive time units. For instance, the 63 SEASAT waveform samplers are spaced over a time span of almost 185 nanoseconds. I will assume that I can use the concept of the "effective" pulse and ignore the actual details of any pulse compression scheme actually used. SEASAT for instance had an effective pulse width of some several nanoseconds, even though the actual signal transmitted from the altimeter was about 3.2 microseconds in duration.

Since the waveform model is viewed in the altimeter's elapsed-time domain, any surface elevations must be converted to elapsed times (or ranging times) through the radar altimeter's two-way ranging time-to-distance factor of $(-c/2)$ where c is the speed of light. The negative sign is included because an increase in surface elevation (the elevation is positive for directions upward out of the sea surface) corresponds to a decrease in the ranging time.

The following description of fundamental assumptions in the convolutional model for rough surface scatter is quoted directly from Brown's paper (ref. 4):

- " 1. The scattering surface may be considered to comprise a sufficiently large number of random independent scattering elements.
2. The surface height statistics are assumed to be constant over the total area illuminated by the radar during construction of the mean return.
3. The scattering is a scalar process with no polarization effects and is frequency independent.
4. The variation of the scattering process with angle of incidence (relative to the normal to the mean surface) is only dependent upon the backscattering cross section per unit scattering area, σ^0 , and the antenna pattern.
5. The total Doppler frequency spread ($4V_r/\lambda$) due to a radial velocity V_r between the radar and any scattering element on the illuminated surface is small relative to the frequency spread of the envelope of the transmitted pulse ($2/T$, where T is the width of the transmitted pulse)."

Over the ocean surface, all of the above assumptions are generally satisfied. However, we must always be careful in selecting the averaging time to insure that surface statistical homogeneity is satisfied. For land scatter, the situation is somewhat different, and some of the above assumptions may be violated.

Under these various assumptions above, the model mean return waveform $W(t)$ is given by the convolution of three terms,

$$W(t) = P_{FS}(t) * q_s(t) * s_r(t) . \quad (15)$$

$P_{FS}(t)$ is the "flat-sea" impulse response function which includes the radar antenna beam-width and pointing angle effects, $q_s(t)$ is the radar-observed surface elevation probability density function, and $s_r(t)$ is the radar altimeter's point-target response function. The convolution of these three terms is represented schematically in Figure 3, and each term will be described in following subsections in this report.

For the SEASAT altimeter the waveform samplers actually measure signal plus noise (ref. 9), so an average noise baseline parameter, b , is needed in a model waveform for fitting to the sampled data. Also, an overall amplitude parameter, A , is needed. Finally the altimeter's altitude tracker moves the entire set of waveform samplers back and forth in time relative to the true position of the mean return waveform, so a tracker-related term $T(t)$ must be included to describe the tracker time-jitter probability density function as well as a possible track-point shift, t_0 . The result of these additional terms is that the final model waveform to be fitted to averages from SEASAT waveform sampler data is given by

$$W(t) = A P_{FS}(t) * q_s(t) * T(t) + b . \quad (16)$$

CONVOLUTION OF:

$$\left(\begin{array}{l} \text{RADAR ALTIMETER} \\ \text{MEAN POWER RETURN} \\ \text{WAVEFORM} \end{array} \right) = \left(\begin{array}{l} \text{FLAT-SEA IMPULSE} \\ \text{RESPONSE (INCLUDES} \\ \text{ANTENNA PATTERN \&} \\ \text{OFF-NADIR ANGLE)} \end{array} \right) * \left(\begin{array}{l} \text{SURFACE ELEVATION} \\ \text{DENSITY FUNCTION} \\ \text{FOR "SCATTERING} \\ \text{ELEMENTS"} \end{array} \right) * \left(\begin{array}{l} \text{RADAR ALTIMETER SYSTEM} \\ \text{POINT-TARGET RESPONSE} \\ \text{(INCLUDES TRANSMITTED} \\ \text{PULSE SHAPE)} \end{array} \right)$$

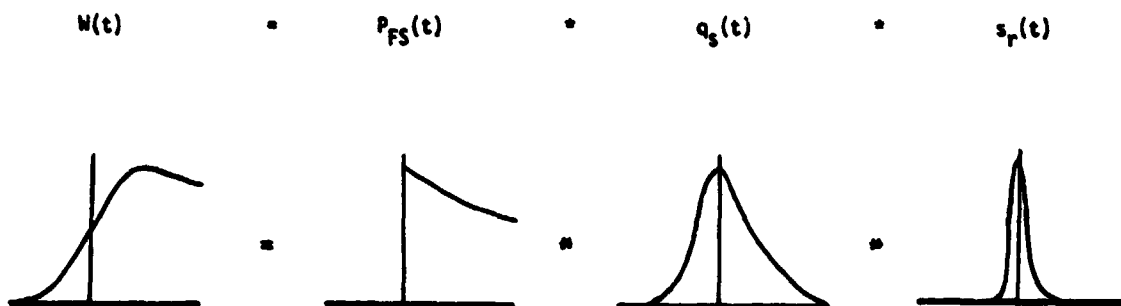


Figure 3. General radar altimeter mean return waveform as convolution of 3 terms.

We have assumed in our SEASAT work to date that the tracker jitter is small enough in width compared to the $q_s(t)$ that we could ignore any part of $T(t)$ except for the possible track-point shift, t_0 . In this case $T(t)$ takes the form of a shifted delta function,

$$T(t) = \delta(t-t_0) . \quad (17)$$

We may find on detailed investigation of the tracker jitter about the true waveform position, that the delta function in $T(t)$ above may need to be replaced by some other form such as a skewed Gaussian whose width and skewness are both functions of the significant waveheight and of the AGC level, but this investigation has not yet been finished.

The final model waveform, from equation (16) above, includes six parameters which are varied in least-squares fitting the input waveform sampler data. These six parameters are listed in the table below. Figure 1 in the Introduction to this report showed qualitatively how the model waveform changes as a result of changes in these six parameters.

In the special case that the $s_r(t)$ and $q_s(t)$ in equation (16) can each be represented by a skewed Gaussian form and that the $T(t)$ has the simple shifted-delta-function form of equation (17), I have described in ref. 5 a several-term analytical expression for $W(t)$; an earlier report of mine (ref. 6) was also based largely on this analytical result. We have more recently decided that, given the SEASAT $s_r(t)$, there is no way to avoid a numerical convolution whenever equation (16) is to be evaluated for any set of values for the parameters in Table II, and we use FFT techniques to speed the numerical convolutions in our computer.

TABLE II. PARAMETERS VARIED IN FITTING MODEL WAVEFORM TO SAMPLER DATA.

<u>Parameter Symbol</u>	<u>Parameter Name or Description</u>	<u>Term Within Eq. (16) in Which Parameter Appears</u>
A	Amplitude	--
t_0	Time-Origin (or track-point)	$T(t)$
σ_s	Sea Surface rms Elevation	$q_s(t)$
b	Average Noise Baseline	--
λ_s	Sea Surface Skewness	$q_s(t)$
ξ	Attitude Angle (or off-nadir angle)	$P_{FS}(t)$

"Flat"-Surface Average Impulse Response Function

This function describes the average power return for a delta-pulse scattered off a "flat" ocean; the quotation marks on flat are to remind us that the sea must be rough at the centimeter scale for the incoherent, rough-surface scatter theory to apply, but that any surface roughness effects at tens of centimeters or greater (i.e., at the oceanographer's level of interest) are described elsewhere in the $q_s(t)$ term. From (ref. 4), the flat-surface average impulse response function is given by

$$P_{FS}(t) = A_0 \exp(-\delta t) I_0(t^{\frac{1}{2}} B) U(t), \quad (18)$$

in which δ is a constant (not the δ -function),

$$\delta = (4/\gamma) (c/h) \cos(2\xi) \quad (19)$$

and

$$\beta = (4/\gamma (c/h)^{\frac{1}{2}} \sin(2\xi) , \quad (20)$$

with $U(t)$ being the unit step function, and $I_0(t^{\frac{1}{2}}\beta)$ being a modified Bessel function. In equations (19) and (20) c is the speed of light, h is the spacecraft altitude, ξ is the absolute off-nadir pointing angle (or attitude angle), and γ is an antenna beamwidth parameter defined as in Brown's equation (ref. 4) by assuming a Gaussian approximation to the antenna gain for an angle θ off the axis,

$$G(\theta) = G_0 \exp[-(2/\gamma)\sin^2\theta] . \quad (21)$$

If θ_w is the usual antenna beamwidth, the angular full-width at 1/2-power points, then

$$\frac{4}{\gamma} = \frac{\ln 4}{\sin^2(\theta_w/2)} . \quad (22)$$

Notice that the time in equation (18) is zero at the instant of the first non-zero power return to the altimeter.

The amplitude term A_0 in equation (18) above contains several other constants:

$$A_0 = \frac{G_0 \lambda_r^2 c \sigma^o(0)}{4(4\pi)^2 L_p^3} \exp(-\frac{4}{\gamma} \sin^2\xi) , \quad (23)$$

where

λ_r is the radar wavelength,

$\sigma^o(0)$ is the ocean surface backscattering cross section at normal incidence,

G_0 is the radar antenna boresight gain, and

L_p is the two-way propagation loss over and above the free-space loss.

In the SEASAT altimeter, the radar return signals are normalized by the automatic gain control (AGC) circuit and we ignore all the individual terms within A_0 in equation (23). Instead A_0 is treated as a simple amplitude scaling constant which is set equal to one for convenience since the A constant in equation (18) will provide the necessary amplitude variation. For future work, note that the combination of fitted A and ξ plus the altimeter's AGC data would lead to a revised σ^o estimate.

Let's emphasize here that the flat surface impulse response function given by equation (18) above is an average-power quantity over many pulses and has no meaning on an individual radar return waveform. Indeed, as Moore and Williams emphasize (ref. 18), the range from a signal level exceeded 5% of the time to a level exceeded 95% of the time is

about 18 dB for the underlying Rayleigh distribution in amplitudes. Any individual pulse is likely to look very different from the mean described in equation (18).

A sample subroutine GTFSR for evaluating the $P_{FS}(t)$ of equation (18) is provided in Appendix D. In evaluating the I_0 in (18), polynomial approximations from Abramowitz and Stegun (ref. 19) are used. In the following $t = z/3.75$.

If $-3.75 \leq z \leq 3.75$, find $I_0(z)$ from

$$I_0(z) = 1 + 3.5156229t^2 + 3.0899424t^4 + 1.2067492t^6 + .2659732t^8 + .0360768t^{10} + .0045813t^{12} \quad (24)$$

If $3.75 < z < \infty$, find $I_0(z)$ from

$$[z^{1/2} \exp(-z)] I_0(z) = .39894228 + .01328592t^{-1} + .00225319t^{-2} - .00157565t^{-3} + .00916281t^{-4} - .02057706t^{-5} + .02635537t^{-6} - .01647633t^{-7} + .00392377t^{-8} \quad (25)$$

For digital computer use, equations (24) and (25) have to be rewritten in nested form to avoid round-off errors from the higher powers in t , and subroutine GTFSR in Appendix D contains the rewritten I_0 polynomial approximations.

Finally, the possibility of a "negative angle" has been built into the flat-sea impulse response function $P_{FS}(t)$ in subroutine GTFSR of Appendix D. The angle ξ is supposed to be a magnitude only and thus a negative angle has no possible physical meaning. Notice that the fastest plateau decay possible in equation (18) is when $\xi = 0$ in which case $\beta = 0$, $\delta = (4/\gamma)(c/h)$, and the $P_{FS}(t)$ becomes

$$P_{FS}(t) = A_0 U(t) \exp(-\delta t) \quad (26)$$

There are, however, experimental situations in which an even more rapid plateau decay is needed. There are at least two such situations: a) the actual ξ is very nearly zero and the noise character of the waveform sampler data will lead to plateau decay rates varying both plus and minus around the decay rate in equation (26) above, or b) erroneous waveform sampler gain calibration data are used so that the decay of the plateau is apparently more rapid than allowed in equation (26).

For this reason, a branch is included in subroutine GTSEA which evaluates equation (18) normally for $\xi \geq 0$ but which evaluates equation (26) with a changing "effective" δ if $\xi < 0$. The magnitude of the change in δ is related to the (negative) magnitude of ξ , and the scaling constant on the δ -change is chosen so that the average of a set of

simulated $\xi = 0$ cases is zero, with individual ξ estimates in this simulation work varying positive and negative.

A negative ξ result from fitting real waveform sampler data is a flag or a warning that there could be waveform modeling errors (or calibration errors); if the ξ estimates are only sometimes negative and are consistent with the expected standard deviation of the ξ -estimation process, then the negative signs pose no problems and simply make the averages of sets of ξ -estimates come out closer to zero. The expected standard deviation of the ξ -estimates can be gotten from a simulation program as sketched later in this report.

Radar-Observed Sea Surface Elevation Probability Density Function

The radar-observed surface elevation probability density function $q_s(t)$ describes the vertical extent and distribution of the ocean surface radar scattering process, and the elevation-to-time transformation is made through the $(-c/2)$ conversion factor. The phrase "radar-observed p.d.f." is used as a reminder that the effective p.d.f. may possibly differ from the geometric surface elevation p.d.f. By geometric surface, I mean the sea surface as seen by optical instruments. Our various waveform analysis work (refs. 6, 7, and 8) has assumed the one specific p.d.f. form given in the next paragraph, but the reader should be aware that there are some alternative p.d.f. forms being proposed. Jackson (ref. 20) proposed one modified $s_r(t)$ form based on theoretical work assuming infinitely long-crested waves, while Lipa and Barrick (ref. 1) have proposed a somewhat different form which includes a "height-slope² cross-skewness" parameter. My approach has been to proceed with one particular often-assumed surface elevation p.d.f. with the possibility of later substituting a different form at such time as the other research on this problem seems sufficiently convincing. Subroutine GTSEA in Appendix D contains the p.d.f. described in the next paragraphs, and the possible future use of different functional forms can be accomplished simply by suitable rewriting of this one subroutine.

The radar-observed surface elevation p.d.f. $q_s(t)$ used here is assumed to be the skewed Gaussian form given in the time domain by

$$q_s(t) = \frac{1}{\sqrt{2\pi} \sigma_s} \left[1 + \frac{\lambda_s}{6} H_3(t/\sigma_s) \right] \exp\left[-\frac{1}{2} (t/\sigma_s)^2\right], \quad (27)$$

where σ_0 is the surface rms waveheight, λ_s is the surface skewness, and H_3 is a Hermite polynomial. The Hermite polynomials for general argument z needed in this paper's discussion are

$$H_3(z) = z^3 - 3z, \quad (28)$$

$$H_4(z) = z^4 - 6z^2 + 3, \quad (29)$$

and

$$H_6(z) = z^6 - 15z^4 + 45z^2 - 15. \quad (30)$$

All satellite altimeter measurements of sea surface roughness have assumed that the significant waveheight (SWH) is exactly four times the rms surface elevation, and since σ_s is in time units, the SWH in distance units is

$$\text{SWH} = 4(c/2)\sigma_s. \quad (31)$$

The λ_s in equation (27) is the surface skewness. Skewness is a non-dimensional quantity, the ratio of a third central moment to the 3/2 power of the second central moment, so no specific use of the $(c/2)$ distance-time conversion factor is needed. It is important to realize however that the sign of the surface skewness in the spatial domain is opposite to the sign in the (ranging) time domain.

The skewed Gaussian form in equation (27) above was used by Pierson and Mehr (ref. 21) in Skylab discussion and by Walsh (ref. 22) in GEOS-3 data as well as in our SEASAT work (refs. 6, 7, and 8); it is a low order case of a general probability function by Longuet-Higgins (ref. 23) for a random variable that is weakly nonlinear. This form is the result of taking the first two terms only in a general Gram-Charlier series (ref. 24). Recent work by Huang and Long (ref. 13) proposed that equation (27) above is not adequate for the surface elevation density and that additional terms must be included in the square of the skewness and in the kurtosis, κ_s , and that the proper surface elevation p.d.f. to use is

$$q_s(t) = \left[1 + \frac{\lambda_s}{6} H_3(t/\sigma_s) + \frac{\kappa_s}{24} H_4(t/\sigma_s) + \frac{\lambda_s^2}{72} H_6(t/\sigma_s) \right] \\ \times \frac{1}{\sqrt{2\pi} \sigma_s} \exp\left[-\frac{1}{2} (t/\sigma_s)^2\right]. \quad (32)$$

We chose to ignore the λ_s^2 and κ_s terms, feeling that these were probably less important in the ocean than in the laboratory. Also the magnitude was likely to be less than the noise in our case. (Figure 6 of Reference 5 shows the effect of the λ_s^2 term for example.) While our work to date has been based on the simpler $q_s(t)$ of equation (27), it is a simple matter to change subroutine GTSEA to the equation (32) form if desired.

There are two reasons for strong interest in the sea surface skewness estimates λ_s . First, this quantity enters some of the theories concerning possible differences between "radar-observed" and "actual" surface elevation p.d.f., the so-called electromagnetic bias in radar altimeters, as in Jackson's paper (ref. 20) for instance. Second, the Huang and Long work (ref. 13) provides the connection between σ_s and λ_s and two other parameters ξ and λ_0 . The significant slope ξ and the dominant wavelength λ_0 are the two important parameters in a new unified two-parameter ocean wave spectral model of Huang et al. (ref. 14), and thus there is a possibility of obtaining surface spectrum information from radar altimeter measurements.

Radar Altimeter Point-Target Response Function

The radar altimeter system point-target response function $s_r(t)$ is the system's effective transmitted pulse as sampled by the waveform sampler set. This point target response is primarily the effective transmitted pulse shape but also includes effects of the receiver bandwidths. The word "effective" above is the verbal rug under which is swept the actual details of pulse compression and implementation.

The only information on the SEASAT $s_r(t)$ is provided by the altimeter Calibration Mode I in which a portion of the transmitted pulse is fed back through the receiver to the waveform sampler set. The SEASAT altitude tracking and SWH estimation in the altimeter hardware were based on an assumed pure Gaussian $s_r(t)$ with a full-width at 1/2-power points of 3.125 nanoseconds. MacArthur (ref. 9) describes a ground-based data processing correction to SWH to allow for the $(\frac{\sin x}{x})^2$ shape which the SEASAT $s_r(t)$ ideally had (as a result of the particular pulse compression technique implemented). I have described in an earlier report (ref. 6) the actual $s_r(t)$ which is not symmetric nor representable by any simple analytical expression. Figure 4 from (ref. 6) sketches this SEASAT $s_r(t)$, based on SEASAT Calibration Mode I Step 9 data, and this figure provides an estimated value every 1.5625 ns. This is the spacing between the five sampling gates at the center of the waveform sampling gate set, while the spacing between all of the rest of the sampling gates is 3.125 ns. We have assumed zero values at every half-gate position between the 3.125 ns-separated samplers, based on the assumed nulls in the ideal $(\frac{\sin x}{x})^2$ response.

In SEASAT the fundamental 1.5625 ns spacing (of the sampled values on all the terms to be numerically convolved) is fixed by the limits on our knowledge of $s_r(t)$ and on its densest sampling being at the 1.5625 ns separation at the center of the sampling gate set. In subroutine FILLV of Appendix C, the subroutine which evaluates the model waveform, the $s_r(t)$ information is carried in array SYS(514) and the two integers NSYS and NSCTR. NSYS is the point after which all remaining SYS values are zero (clearly NSYS < 514), and NSCTR

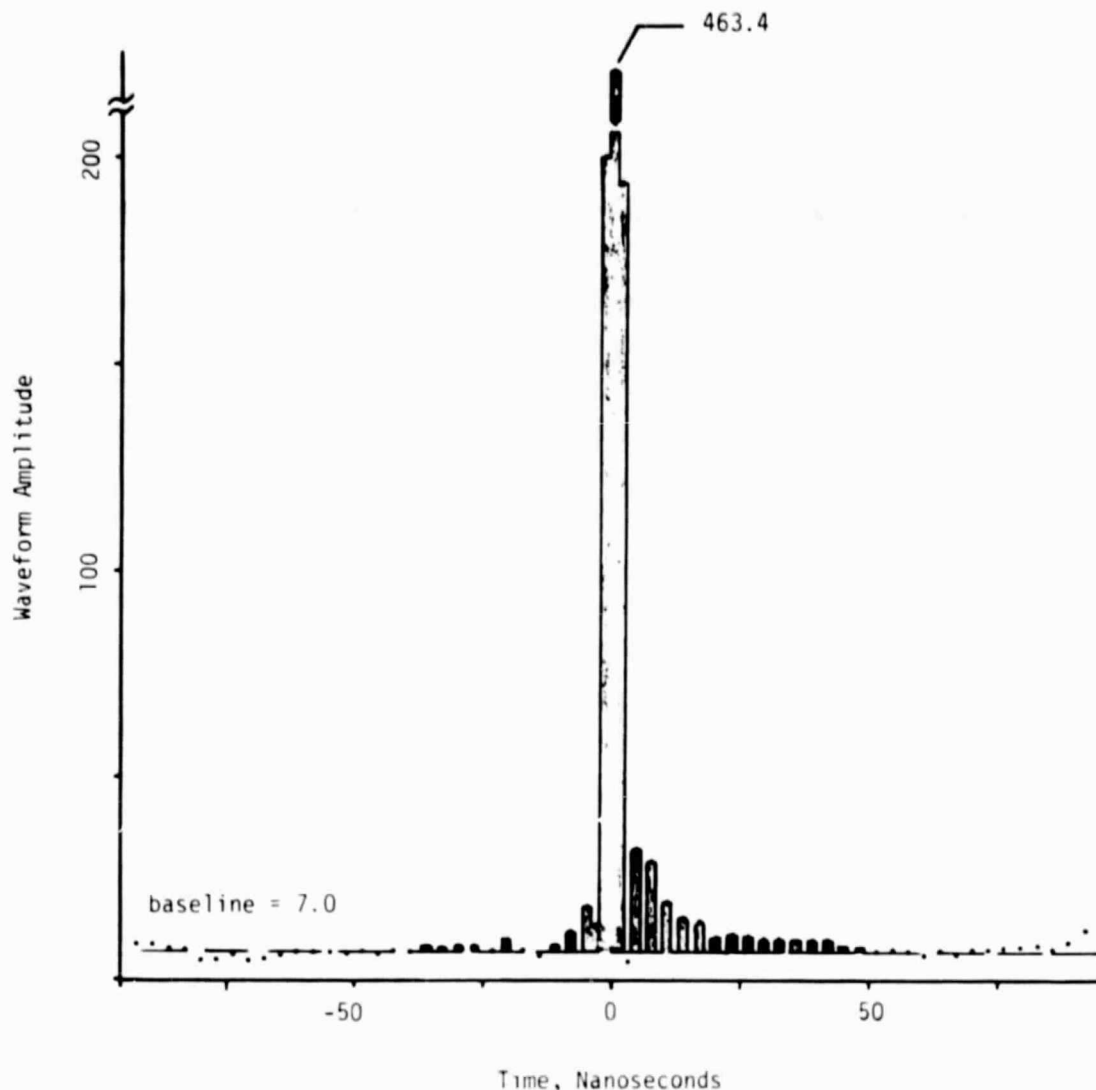


Figure 4. SEASAT radar altimeter system point-target response from CAL Step 9 after gain bias corrections.

is the index of the peak or center at the sampled $s_r(t)$. NSCTR is used in a time shifting step which in effect sets the peak element within SYS at the time zero.

If one can assume that the $s_r(t)$ has a skewed Gaussian form like equations (27) or (32) of the preceeding subsection, then my Reference 5 provides a several-term analytic expression for the model waveform and subroutine FILLV of Appendix C could be replaced by a coded version of that expression. This may be useful for simulations of future altimeters, but I expect that any actual altimeter will have a $s_r(t)$ at least as complicated as the SEASAT case in Figure 4, and that the full convolution procedure of this present report will again be necessary.

The preceding discussion in this section on model waveforms has emphasized the need to carry out a numerical convolution because of the actual SEASAT $s_p(t)$ form. Moreover, my discussion of $q_s(t)$ has at least implied the desirability of being able to change the functional form used for this surface elevation p.d.f. This in turn means at least two numerical convolutions for a set of model waveform values. Then the step-estimation of derivatives within the fitting process will require four more model waveform sets to be generated if six parameters are fitted (in general, six parameters being fitted would require six more model waveform sets to be generated, but in this problem we know already the derivatives of equation (16) with respect to the parameters A and b). Finally the general fitting program will require from five to ten iterations through the waveform-plus-derivative loop. In brief, there are many convolution steps required to estimate waveform parameters for one set of averaged waveform sampler data, and ways to speed this processing must be found.

One way to speed the convolution is to take advantage of the convolution theorem in Fourier transform theory. This theorem asserts that if one has two time functions $p(t)$ and $q(t)$ whose Fourier transforms $P(f)$ and $Q(f)$ exist, the Fourier transform of the convolution $p(t)*q(t)$ is the product $P(f)Q(f)$. Consequently, the convolution $p(t)*q(t)$ can be performed by: a) taking the Fourier transforms of $p(t)$ and $q(t)$ to obtain $P(f)$ and $Q(f)$; b) forming the (complex) product $[P(f)Q(f)]$; and c) taking the inverse Fourier transform of $[P(f)Q(f)]$ to obtain the desired answer. This procedure also can be used for two sampled time functions using the discrete Fourier transform (DFT) and the inverse DFT, providing that the two functions can be set to zero outside a region of interest. The computational efficiencies of the Fast Fourier Transform (FFT) result in the above-sketched convolution through transforms technique being faster in many cases than the simpler straightforward numerical convolution, and we have found by trying both methods that there is a speed advantage to convolution by transforms within the model waveform computation.

There is a large body of literature concerning Fourier transforms, DFTs, FFTs, convolution, and so forth; the textbook by Brigham (ref. 12) is a source which is quite readable and accessible, and upon which the discussion in these paragraphs is based. The particular FFT and inverse FFT programs used here for model waveform computations were the subroutines FFA and FFS and their associated subroutines from a published package (ref. 25) of digital signal processing programs. FFA is a radix 8-4-2 FFT routine for a real data sequence which performs as many base 8 iterations as possible and then performs one base 4 or base 2 iteration if necessary, and FFS is the radix 8-4-2 inverse counterpart of FFA. The inclusion of the radix 8 steps in FFA and FFS leads to longer Fortran source code but shorter computation times than for other simpler FFT routines, and additional timing discussion is provided by Reference 25.

The FFT-convolution will require that the total array length N_T , used in each term to be convolved, be the same and in general there is the FFT requirement that N_T be some power of 2, $N_T = 2^M$. The smaller the value of M , the faster the computation, but M must be large enough to avoid overlap effects as discussed in ref. 12. Suppose again that two (discrete) sequences $p(t)$ and $q(t)$ are to be convolved and that $p(t)$ has N_p non-zero sampled values while $q(t)$ has N_q such values. There will be $(N_T - N_p)$ and $(N_T - N_q)$ successive zero values in the arrays for $p(t)$ and $q(t)$ respectively. Then the requirement that $N_T \geq N_p + N_q - 1$ will avoid the overlap problem. If there are three sequences $p(t)$, $q(t)$, and $s(t)$ with numbers N_p , N_q and N_s , the requirement on N_T becomes $N_T \geq N_p + N_q + N_s - 2$. Obviously we are assuming the same sample time intervals Δ for $p(t)$, $q(t)$, $s(t)$ and for the final answer.

In the specific SEASAT waveform problem, the sampling time Δ_s is 1.5625 ns, the spacing of the gates in the center of the set of waveform samplers. The total width spanned by the sampling gates is around 120 Δ_s and the flat sea response function $P_{FS}(t)$ must be allowed at least this width of non-zero values so $N_p \approx 120$. To allow for up to 20 m SWH implies that upwards of 200 Δ_s is needed to represent $q_s(t)$ out to the $\pm 4\sigma_s$ level, so $N_q \approx 200$. Then the system point-target response function has a width upwards of 100 Δ_s , so $N_p \approx 100$. This leads to a total $N_T = 120 + 200 + 100 = 400$, and the next highest 2^M is at $M = 9$ for $N_T = 512$. The actual widths used vary somewhat from this illustrative example, but $N_T = 512$ is the total array length used.

The model waveform evaluation subroutine FILLV in Appendix C uses 512-point arrays for the $P_{FS}(t)$, $q_s(t)$ and $s_r(t)$ and 512-point FFT evaluation (the actual array dimensions are 514 because of the details of the FFA and FFS subroutines used). The time-shift term $T(t)$ of equation (17) is not written as a separate term in the convolution, but is accomplished through the Fourier transform time-shift theorem (ref. 12); a shift in the time domain leads to a phase shift in the frequency domain, so that for example if $p(t)$ and $P(f)$ are Fourier transform pairs, then the Fourier transform of $p(t - t_0)$ is $P(f) \exp(-j 2\pi f t_0)$. A similar property applies to the discrete Fourier transform, and FILLV takes advantage of this and includes the phase shift in the complex product in the transform domain before doing the inverse FFT to bring the model waveform back to the time domain. Details of this are obvious in the FILLV source code in Appendix C.

Finally, the convolution via FFT method described above was tried initially as an experiment, to see if the computation speed was improved over the straightforward convolution of two time functions; there was an improvement of nearly an order of magnitude, so the FFT method was kept in our waveform fitting. Adding one more convolution for a total of three terms convolved then became simple and cost less than twice the time for two terms only to be convolved via FFTs, and this allowed us to separate the problem so that $P_{FS}(t)$, $q_s(t)$, and $s_r(t)$ were entered by different subroutines. This modular structure will easily permit future possible changes in the functional form used for $q_s(t)$. Also as

we note later, most of the time in our waveform fitting computations is taken by the computer doing FFTs, and this makes attractive the process of using one of the available array processors in any future computer system which spends an appreciable part of its time doing waveform fitting for modeled parameter recovery.

ADDITIONAL PROCESSING CONSIDERATIONS

Use of Calibration Data

If the set of SEASAT waveform sampler measurements is Y_i , $i = 1$ to 63, it has been assumed that there is a set of gain corrections g_i and offset (or shift) corrections s_i such that a corrected set of sampler values Y_i is obtained from

$$Y_i = g_i(Y_i - s_i), \quad i = 1 \text{ to } 63. \quad (33)$$

The offsets s_i were to be obtained from the altimeter's standby mode, under the assumption that they were generated somewhere in the sampler-to-telemetry system interface. The gains g_i were to be obtained from the altimeter's calibration mode II in which a uniform signal level is presented to the input of the waveform sampler set so that the output differences can be interpreted as gain variations from which a set of gain correction factors can be obtained. We have already described (ref. 6) some of the problems with the gains g_i derived from Calibrate Mode II of the altimeter and, more recently, we found that there was an apparent gain change correlated with whether the altimeter's AGC word is above or below 32 dB. The SEASAT altimeter's Calibration Mode I supplies the sampled point-target response function $s_r(t)$ but even here the sampler gain calibration problems cause some concern.

There is work yet to finish on the SEASAT sampler gain problems, and no final values for g_i and s_i in equation (33) can be given here. For purposes of this report, it will be simply assumed that any necessary waveform sampler calibrations have already been applied to the averaged sample data prior to the radar altimeter waveform modeled parameter recovery procedure.

Separate Subroutines for Terms Within Waveform Convolution

I want to emphasize here the desirability of keeping separate waveform terms in separate subroutines because of the freedom and flexibility in then changing any one of

these terms. Once there is a numerical convolution step required in a model waveform (and I cannot imagine that this will change in future altimeter systems), then the use of an FFT-convolution process means that adding an additional convolution will not even double the processing time. For example, Reference 5 gives the answer for the convolution of the flat-sea impulse response function $P_{FS}(t)$ and the surface elevation density function $q_s(t)$ for the particular forms of $P_{FS}(t)$ and $q_s(t)$ which I am now using. Yet I prefer to do this particular convolution numerically too (through FFT again) in order to preserve for future work the ability to try different functional forms for $q_s(t)$, forms which would invalidate the results from Reference 5.

Computer Running Time Comments

Table III below lists typical computation time taken by the general six-parameter estimating iterative least-squares waveform fitting program on the Data General ECLIPSE S-200 computer system in Building E-106 at Wallops Flight Center for three different methods of generating the model waveform and its derivatives.

TABLE III. COMPARISON OF TYPICAL SEASAT WAVEFORM FIT TIMES ON S-200 COMPUTER

<u>Method</u>	<u>Waveform Generation</u>	<u>Derivative Generation</u>	<u>Waveform Fit Times</u>
"Full Analytical"	Analytical, from Ref. [5]	Analytical derivatives of Ref. [5] expressions	1-2 seconds
"Analytical Numerical"	$[P_{FS}(t) * q_s(t)]$ Analytical; then Simple Numerical Convolution with $s_r(t)$	Step-Change in Each Parameter, Reevaluate, for $\left[\frac{\Delta \text{Waveform}}{\Delta \text{Parameter}} \right]$	2-3 minutes
"Full Numerical by FFT"	FFT for $P_{FS}(t), q_s(t)$ $s_r(t)$, Then Complex Product and Inverse FFT	Step-Change ...	15-25 seconds

In planning future altimeter data processing systems, the 20 seconds per fit of the "Full Numerical by FFT" entry in the table above would seem to pose a problem since the waveform data will probably be averaged and waveform fits done for typically every 2.5 seconds of data. The factor of 8 between processing time and real-time would create an

intolerable data processing bottleneck, but the existence of add-on hardware array processors (and their associated software) suggests a way out of the problem. In Table III, the additional time of the "Full Numerical by FFT" compared to the "Full Analytical" is primarily spent doing FFTs in the Fortran programs. Let's assume that all the rest of the waveform fitting processing time could be gotten below one second. This is probably a reasonable goal given the relative clumsiness of the Fortran source code written for the waveform fitting programs which were developed as research tools and not as production-oriented programs. Then let's estimate how many FFT and inverse FFT operations there are in one typical waveform fit, and estimate the operating time required by one representative array processor (ref. 26) which can be interfaced to Data General computers such as the ECLIPSE S-200 on which Table III is based.

To evaluate a model waveform, subroutine FILLV in Appendix C does two FFTs (on $P_{FS}(t)$ and $q_s(t)$, with transform of $s_r(t)$ already having been performed once in the program initialization), then three complex multiplications (multiplication of the transforms of $P_{FS}(t)$, $q_s(t)$, and $s_r(t)$ and of the $T(t)$ -generated phase shift), and then one inverse FFT. Thus in effect three FFTs and three complex multiplications of 512-point array quantities are required to generate one set of model waveform samples. The MSP-2 array processor from Computer Design and Applications, Inc. (ref. 26) does a 512-point real FFT in 3.26 milliseconds and a 512-point complex product in 1.28 ms so that the total time per model waveform is about $3(3.26) + 3(1.28)$ ms or about 13.6 ms. Then the derivatives for the six parameters fitted will require generation of four more model waveforms for step changes in different parameters in subroutine FILLD in Appendix C, so that one pass through the iteration loop in the least-squares fitting program uses a total of five waveform-generating steps or $5(13.6) = 68$ ms. Then the average number of iterations in the waveform fitting process is generally eight or less so that the total waveform generating time in one complete waveform fit is around $8(68) = 550$ milliseconds. If the array sizes discussed were to increase to 1024 instead of 512 this time would become about 1.16 seconds, slightly more than twice 550 milliseconds, based again on the MSP-2 processor literature (ref. 26).

Thus the entire waveform fitting procedure for one set of waveform sampler averages could probably be carried out in under two seconds of computer time on an ECLIPSE S-200 computer with an array processor. I use the Computer Design and Applications MSP-2 array processor only as an example to show that a 2.5 second average of waveform data could be processed in less than 2.5 seconds of computation. Other array processors are available for minicomputer or large mainframe computer systems at varying prices, speeds, and features. At least one unit available for ECLIPSE computers runs more than twice as fast as the MSP-2 processor in the example above. No array set-up or transfer times or other system overhead times have been included in the above examples, and the possible use of an array processor deserves a more detailed study for future altimeter waveform proc-

essing. Finally, it should be noted that the computation times required by many of the current 32-bit "super-minicomputers" is considerably less than the computation time of our ECLIPSE S-200 system; it might be possible that a newer computer would operate fast enough that no array processor is needed, but this again is a further issue for detailed study.

Simulation Program for Assessing Operation of Fitting Program

The model waveform generating procedure has been discussed earlier. Verification of correct operation of this program is fairly simple; one generates a waveform for a given parameter set and checks that the result matches what the theory predicts. One way to test the SEASAT waveform program FILLV of Appendix C is to write a small test program to produce a skewed Gaussian characterized by a rms σ_s and λ_s and to then take 1.5625 ns-separated samples of this function to form a point-target function $s_p(t)$ in array SYS; then the waveform generated by FILLV for these SYS must agree closely with values calculated from the method described by Reference 5. This has been done for the SEASAT model waveform used here and the verification was satisfactory.

Another level of checking is needed as well to verify that the waveform least-squares fitting program recovers the correct parameters from the input waveform samples. For this check, a simulation program was developed which generates model waveform samples for one particular set of waveform parameters and then uses those sample values as input to the waveform fitting program; a comparison of the parameters recovered and the input parameters will then reveal any errors or biases induced by the fitting program. This simulation program is also extremely useful in determining the fundamental limits to parameter recoverability, limits due to the fundamental noise-like character of the individual radar return waveform.

Figure 5 sketches the simulation program used here to investigate waveform parameter recoverability. The average of N individual returns, for any particular specified time-point on the waveform, is expected to have a Gaussian distribution because of the central limit theorem. The individual return itself is described by an exponential distribution whose mean and standard deviation are equal, so the standard deviation of the N-average return is expected to have a standard deviation equal to the mean divided by \sqrt{N} , and this noise modeling is built into the "noise loop" shown in Figure 5. The results of considerable experimenting with this simulation for the SEASAT work showed that the waveform fitting program operated with no appreciable biases and appeared tolerant of a reasonable range of first-guess parameter values. No further details will be presented here as this is the subject for a future separate report. The important point for this report is that this sort of simulation program must be produced and exercised for each different radar altimeter's particular conditions, and that work on waveform processing software is not

complete until the simulation program has exercised the waveform modeled parameter recovery programs.

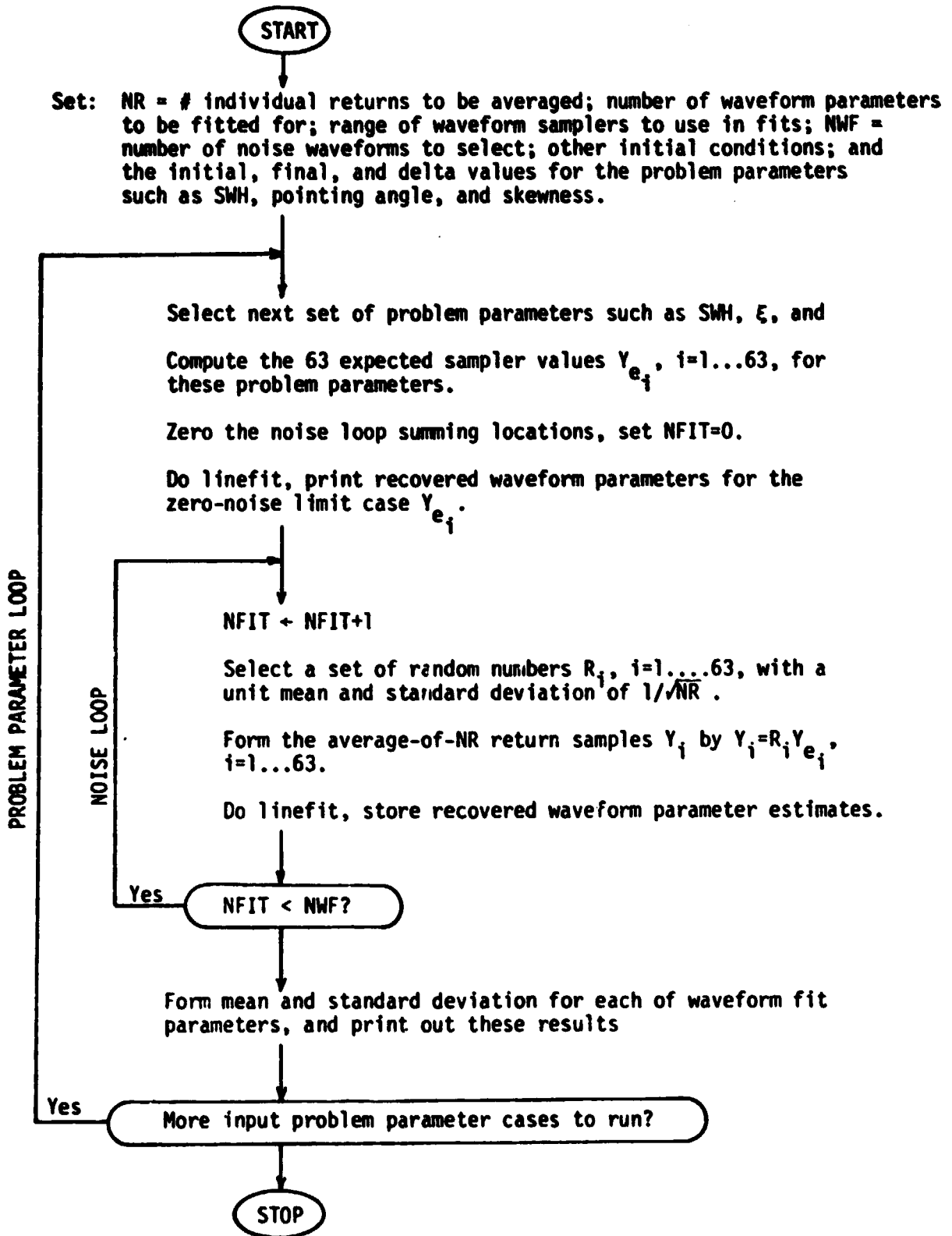


Figure 5. Simulation program for assessing waveform parameter recoverability.

REFERENCES

1. Lipa, B. J.; and Barrick, D. E.: "Ocean Surface Height-Slope Probability Density Function from SEASAT Altimeter Echo," Paper submitted to J. Geophys. Res. SEASAT Special Issue, 1981.
2. Hayne, G. S.: "Initial Development of a Method of Significant Waveheight Estimation for GEOS-III," Applied Science Associates, Inc., NASA Contract NAS6-2650, NASA CR-141425, 1977.
3. Fedor, L. S.; Godbey, T. W.; Gower, J. F. R.; Guptill, R.; Hayne, G. S.; Ruferach, C. L.; and Walsh, E. J.: "Satellite Altimeter Measurements of Sea State: An Algorithm Comparison," J. Geophys. Res., Vol. 84, No. B8, pp. 3991-4001, 1979.
4. Brown, G. S.: "The Average Impulse Response of a Rough Surface and Its Applications," IEEE Trans. Antennas and Propagat., Vol. AP-25, No. 1, January 1977, pp. 67-74. Also in IEEE Trans. Oceanic Eng., Vol. OE-2, No. 1, January 1977, pp. 67-74.
5. Hayne, G. S.: "Radar Altimeter Mean Return Waveforms from Near-Normal-Incidence Ocean Surface Scattering," IEEE Trans. Antennas and Propagat., Vol. AP-28, No. 5, September 1980, pp. 687-692.
6. Hayne, G. S.: "Wallops Waveform Analysis of SEASAT-1 Radar Altimeter Data," Applied Science Associates, Inc., NASA Contract NAS6-2810, NASA CR-156869, 1980.
7. Hayne, G. S.; Hancock, D. W., III; Huang, N. E.; Walsh, E. J.; and Forsythe, R. G.: "Determination of Ocean Significant Slope and Dominant Wavelength by SEASAT-1 Altimeter Waveform Analysis," Paper presented at the SEASAT Storms Colloquium, University of Miami, Miami, Florida, October 29-31, 1980.
8. Hayne, G. S.; and Hancock, D. W., III: "Sea-State-Related Altitude Errors in the SEASAT-1 Radar Altimeter," Paper submitted to J. Geophys. Res. SEASAT Special Issue, 1981.
9. MacArthur, J. L.: "SEASAT-A Radar Altimeter Design Description," SDO-5232, Applied Physics Laboratory of The Johns Hopkins University, November 1978.
10. Townsend, W. F.: "An Initial Assessment of the Performance Achieved by the SEASAT-1 Radar Altimeter," IEEE J. Oceanic Eng., Vol. OE-5, No. 2, pp. 80-92, 1980.
11. Hancock, D. W.; Forsythe, R. G.; and Lorell, J.: "SEASAT Altimeter Sensor File Algorithms," IEEE J. Oceanic Eng., Vol. OE-5, No. 2, pp. 93-99, 1980.
12. Brigham, E. Oran: The Fast Fourier Transform, Prentice-Hall, Inc., 1974.
13. Huang, N. E.; and Long, S. R.: "An Experimental Study of the Surface Elevation Probability Distribution and Statistics of Wind Generated Waves," J. Fluid Mech., Vol. 81, pp. 179-200, 1980.
14. Huang, N. E.; Long, S. R.; Tung, C. C.; Yuen, Y.; and Bliven, L. F.: "A Unified Two-Parameter Wave Spectral Model for a General Sea State," J. Fluid Mech., 1981.

15. McWilliams, P.; Hall, W. S.; and Wagner, H. E.: "Multichannel Analyzer Data Analysis by a Least-Squares Computer Program," Rev. Sci. Instr., Vol. 33, No. 1, pp. 70-73, January 1962.
16. McKinnon, A. E.; Szabo, A. G.; and Miller, D. R.: "The Deconvolution of Photoluminescence Data," J. Phys. Chemistry, Vol. 81, No. 16, pp. 1564-1570, 1977.
17. Niple, E.: "Nonlinear Least-Squares Analysis of Atmospheric Absorption Spectra," Applied Optics, Vol. 19, No. 20, pp. 3481-3490, 1980.
18. Moore, R. K.; and Williams, G. S., Jr.: "Radar Terrain Return at Near Vertical Incidence," Proc. IRE, Vol. 45, No. 2, February 1957, pp. 228-238.
19. Abramowitz, A.; and Stegun, I. A.; eds.: NBS Handbook of Mathematical Functions, Dover Books, Inc., 1972.
20. Jackson, F. C.: "The Reflection of Impulses from a Nonlinear Random Sea," J. Geophys. Res., Vol. 84, pp. 4939-4943, 1979.
21. Pierson, W. J.; and Mehr, E.: "Average Return Pulse Form and Bias for the S193 Radar on Skylab as a Function of Wave Conditions," in The Use of Artificial Satellites for Geodesy, S. W. Henrickson et al.; eds., Geophysical Monograph Series, Vol. 15, American Geophysical Union, pp. 217-226, 1972.
22. Walsh, E. J.: "Extraction of Ocean Waveheight and Dominant Wavelength from GEOS-3 Data," J. Geophys. Res., Vol. 84, pp. 4003-4010, July 1979.
23. Longuet-Higgins, M. S.: "The Effect of Nonlinearities on Statistical Distributions in the Theory of Sea Waves," J. Fluid Mech., Vol. 17, pp. 459-480, 1963.
24. Kendall, M. G.; and Stuart, A.: "The Advanced Theory of Statistics," Vol. 1: Distribution Theory, Second ed. Charles Griffin & Co., Ltd., (London), 1963.
25. IEEE Acoustics, Speech and Signal Processing Society, Programs for Digital Signal Processing, Digital Signal Processing Committee, eds., IEEE Press, 1979.
26. MSP-2 Micro Signal Processor Product Information, and MSP-2 Peripheral Array Signal Processor Operation and Maintenance Manual, Computer Design and Applications, Inc., 1980.

APPENDIX A

GENERAL DISCUSSION RELATED TO WAVEFORM FITTING SUBROUTINES

The following four appendixes in this report provide source listings for various subroutines in the waveform fitting procedures and for SEASAT-1 data. The main program, which is not provided here, handles obvious details of data input, averaging, and calibration. There are also some initial set-up details handled in the main program, setting the number of parameters to be fitted, lower and upper sampling gate limits if fewer than 63 waveform sampler results are to be used, setting the array for the system point-target response, and so on. The main program also sets the values of array SYS which carries the sampled system point-target response function $s_p(t)$, and sets the values of an array GUESS which carries first parameter guess for each parameter being determined in the iterative fit within SWHFIT. The array also carries the fixed values of the parameters not being fitted if fewer than six parameters are being determined.

Table A-1 lists subroutines used in the waveform fitting and indicates where their source listings are to be found. Table A-2 shows which of the subroutines are called by others; obviously subroutine SWHFIT is called from the main program. In the Data General FORTRAN used, "X" in column one of a source line means the line is to be ignored unless the /X switch is used in the FORTRAN compile step. Various debug statements in the source listings, with X in column one, call subroutine RLOUT and CPOUT which are not supplied. RLOUT and CPOUT are trivial array formatting and printout routines for real arrays and complex arrays respectively.

The calling list for SWHFIT includes the 63 waveform sampler corrected data averages in YIN(63), and sampling gate numbers NLO and NUP. Data for gates from NLO to NUP will be fitted, and normally NLO=1 and NUP=63. The calling list array WTY(63) carries relative weighting factors and, for the generally-used uniform data weighting all the elements of WTY are set to unity. Notice the array STPRM(7) which is set to the step sizes to be taken in producing the parameter derivative estimates by FILLD. Actually the first, fourth, and seventh values of STPRM have no meaning since the amplitude (first) and baseline (fourth) derivatives are found without making a step-change in parameters one and four, and the seventh value of STPRM was intended for the kurtosis which has not (to date) been written into the function in GTSEA. Notice also in the SWHFIT comments the description of JORDR(7), an integer array determining the order in which the parameters are to be found.

Part of the data communication in these subroutines is by calling lists and part by named COMMON areas. Table A-3 lists the contents of named COMMON area SSM4N, and Table

A-4 lists the contents of named COMMON area SYSTM. Two additional named common areas CON and CON1 are used within the FFT package only.

The individual subroutines have a number of additional comments provided in their source listings. It is hoped that these comments, the Tables A-1 through A-4, the remarks above, and the the main body of this report should provide the reader an adequate acquaintance with the Wallops waveform modeled parameter recovery procedures.

TABLE A-1. SUBROUTINES USED IN WAVEFORM MODELED PARAMETER RECOVERY PROGRAMS FOR SEASAT-1 DATA

<u>Routine Name</u>	<u>Where Provided</u>	<u>Purpose and Descriptive Remarks</u>
SMHFIT	Appendix B	Function subroutine to fit up-to-6-parameter model waveform to corrected averaged waveform sampler data supplied by main calling program.
SYMINV	Appendix B	Symmetric matrix inversion subroutine used by SMHFIT.
FILLV	Appendix C	For given parameter set, generates model waveform at the 63 waveform sampler locations. Waveform is generated through FFT-numerical convolution of three separate terms.
FILLD	Appendix C	Uses FILLV, changes each parameter by specified step to form estimates of derivatives of model waveform with respect to each of the waveform fit parameters.
GTFSR	Appendix D	Produces flat-sea impulse response function, one of the terms convolved in FILLV.
GTSEA	Appendix D	Produces radar-observed surface elevation p.d.f., one of the terms convolved in FILLV.
FFA	Appendix E	Does Fourier transforms in radix 2-4-8 FFT package used.
FFS	Appendix E	Does inverse Fourier transforms in radix 2-4-8 FFT package used.
ORD1, ORD2, R2TR, R4SYN, R4TR, R8SYN, R8TR	Not Provided	Additional subroutines in radix 2-4-8 FFT package not provided in this report but available from Reference [25] or as described in the comments in FFA and FFS in Appendix E.
RLOUT, CPOUT	Not provided	Routines for printing out real and complex arrays, respectively, and used for debugging only.

TABLE A-2. RELATIONSHIPS AMONG WAVEFORM FITTING SUBROUTINES

Subroutines Called from Routines in Column at Left Side of this Table

Waveform Fitting Subroutines	SYMINV	FILLV	FILLD	GTFSR	GTSEA	FFA	FFS	ORD1	ORD2	R2TR	R4SYN	R4TR	R8SYN	R8TR	RLOUT	CPOUT
SMHFIT	✓	✓	✓												*	*
FILLV				✓	✓	✓	✓								*	*
FILLD		✓													*	
FFA								✓	✓	✓	✓	✓		✓		
FFS								✓	✓	✓	✓	✓	✓			

Notes: ✓ = Always Needed.

* = Debug (optional) Routines RLOUT, or CPOUT to Print Values in Real or Complex Arrays Respectively.

TABLE A-3. DESCRIPTION OF VARIABLES IN NAMED COMMON AREA /SSM4N/

<u>Variable</u>	<u>Description</u>
A(7)	The seven final values of the up-to-seven parameters fitted to input waveform data. NA = number fitted. (The seventh is kurtosis, not yet written into GTSEA, so A(7) = 0 always.)
AEDIT(2,7)	Provides for possible lower, upper edit limits for the A(7).
CALM	"Calm-Sea Constant" or effective 1- σ point-target response width used in finding SWH with "analytical" fitting. When numerical convolution (via FFT) is used, main program should set CALM = 0.
CNSTR(7)	Constraints on the fitted A(7) which in effect limit possible size of parameter changes in the fitting iterations. CNSTR values are set by <u>a priori</u> information, once, at start of main program.
CORRL(21)	Provides information on correlation of fitted parameters - probably of doubtful use in this non-linear problem.
GUESS(7)	Carries first-guess for waveform parameters to start iterative fit procedure, and carries fixed values for parameters <u>not</u> being fitted.
ITER	Indicates # iterations needed to fit data upon return from SWHFIT.
NA	Number of the A(7) being fitted, with maximum NA = 7 (if kurtosis is added, for example - current maximum NA is 6).
JORDR(7)	Set by main program, indicates order of parameters being fitted, and provides ability to fit different parameter combinations without rewriting source code.
SERSQ	Sum of squares of fit residuals, upon return from SWHFIT.
T(63)	Carries the 63 independent parameter (sampler time) values. In SEASAT, T(1) = -92.1875 and T(63) = 92.1875.
XCNST(7)	Provided for transporting different constants, but not used for anything significant at this time.

TABLE A-4. DESCRIPTION OF VARIABLES IN NAMED COMMON AREA /SYSTM/

<u>Variable</u>	<u>Description</u>
NSYS	Value of index I after which all SYS(I) are zero.
NSCTR	Center-of-response index, the index within SYS to which is attached the time zero label.
SYS(514)	Contains the sampled point-target response in the elements SYS(1) through SYS(NSYS), with all remaining elements through SYS(514) zero <u>before</u> the first call to FILLV. After that, SYS contains the 512-point FFT transform of the input $s_r(t)$ and must not be changed.
SYSID(21)	A comment line carrying information on how SYS was determined. SYSID is only of interest to the main program.

APPENDIX B.

SOURCE LISTINGS FOR SUBROUTINES SWHFIT AND SYMINV

```

0001 C DPB:HAYNE:SWHFIT3.FR REVISED 10/30/80, APPROX. 1315 HRS
0002 C
0003 C G.S.HAYNE WFC/DAS 09/25/80
0004 C
0005 C SUBROUTINE SWHFIT3 IS EXACTLY SAME AS SWHFIT2 EXCEPT THAT
0006 C A NEW COMMON AREA /SYSTEM/ HAS BEEN ADDED AND THE COMMUNICATION
0007 C WITH FILLV & FILLD CHANGED SLIGHTLY. THIS IS SO THAT THE
0008 C SYSTEM POINT-TARGET RESPONSE DATA FROM FILLV IS AVAILABLE
0009 C BACK AT THE MAIN CALLING PROGRAM.
0010 C
0011 C SMALL CHANGE ON 10/30/80 CHANGED WHAT HAPPENED FOR A(3)
0012 C NEAR ZERO.
0013 C
0014 C CHANGE ON 07/16/80 ADDED ARGUMENT IXDBG TO ROUTINE CALLING
0015 C LIST; A VALUE OF 1 FOR IXDBG SIGNALS PRINTOUT OF EACH
0016 C INDIVIDUAL ITERATION IN THE FIT.
0017 C
0018 C THIS VERSION FOR THE 512-POINT TRANSFORM, AND COMMON AREA
0019 C SSM4N.
0020 C
0021 C THIS IS REVISION OF SWHSSN (REV.07/27/79) NOW INCORPORATING
0022 C FFT-CONVOLUTION METHOD OF GENERATING SEASAT-1 FUNCTION VALUES
0023 C AND DERIVATIVES. THE SUBROUTINES NEEDED BY SWHFIT INCLUDE:
0024 C FILLV3,FILLD3,FAST,FSST,FR2TR,FR4TR,FR4SYN,FORD1,
0025 C FORD2,GTSEA2,GTFSR2, AND SYMINV.
0026 C NLO & NUP ARE NOW IN CALLING LIST OF SWHFIT; THESE DEFINE
0027 C LOWER AND THE UPPER LIMITS TO THE NUMBER OF POINTS BEING
0028 C FITTED. IT IS REQUIRED THAT NLO.GE.1, NUP.LE.63, AND THAT
0029 C NA.LT.(NUP-NLO).
0030 C
0031 C
0032 C AS IN SWHSSN, THE INTEGER ARRAY JORDR(7) DESCRIBES THE ORDER
0033 C IN WHICH PARAMETERS ARE TO BE FITTED. THE FIRST NA (NA.LE.7)
0034 C ELEMENTS OF JORDR(.) ARE TO BE INDIVIDUAL (AND DIFFERENT) VALUES
0035 C OF J, WHERE J HAS THE FOLLOWING MEANING:
0036 C J = 1 DESIGNATES AMPLITUDE
0037 C " 2 " TIME ORIGIN
0038 C " 3 " RISE TIME
0039 C " 4 " BASELINE
0040 C " 5 " SKEWNESS
0041 C " 6 " OFF-NADIR ANGLE
0042 C " 7 " KURTOSIS.
0043 C
0044 C
0045 C A 7-PARAMETER NON-LINEAR LEAST-SQUARES FITTING FUNCTION SUB-
0046 C ROUTINE SPECIALIZED TO CASE OF SEASAT SIGNIFICANT WAVEHEIGHT
0047 C DETERMINATION. SUBROUTINE SWHFIT INCLUDES POSSIBILITY OF UNEQUAL
0048 C WEIGHTS FOR THE 63 INPUT S&H DATA POINTS. WEIGHTING BY INVERSE
0049 C OF VARIANCE ESTIMATES, FOR EXAMPLE, PRODUCES IN EFFECT A
0050 C MAXIMUM-LIKELIHOOD FIT. THE INDICATOR IER IS 1 ON THE RETURN
0051 C FROM A SUCCESSFUL FIT; OTHERWISE IER CAN INDICATE THE TYPE OF
0052 C ERROR AS INDICATED IN COMMENTS BELOW.
0053 C
0054 C
0055 C THERE'S ALSO PROVISION FOR CALCULATING CORRELATION COEFFICIENTS,
0056 C AND ABILITY TO ADD A PRIORI FIT CONSTRAINTS, RESULTING FROM
0057 C CONVERSATIONS WITH BILL WELLS OF MASCI, INC.
0058 C

```

```

0059 C
0060 C      SUBROUTINE SWHFIT USES SYMMETRIC MATRIX INVERSION ROUTINE SYMINV
0061 C      TO INVERT SYMMETRIC MATRIX XMAT IN PROBLEM. FUNCTIONAL FORM TO
0062 C      BE FITTED IS SUPPLIED BY SUBROUTINE FILLV AND THE NEEDED
0063 C      DERIVATIVES BY SUBROUTINE FILLD.
0064 C
0065 C      FUNCTION SWHFIT(NLO,NUP,VIN,WTY,IXDBG,IER)
0066 C      COMPILER STATIC
0067 C
0068 C      REAL STPRM(7)/1.0, 0.25, .20, 1.0, 0.05, 0.05, 0.05/
0069 C
0070 C      DIMENSION VIN(63),WTY(63),XMAT(7,7),BCOLM(7),PVECT(7),
0071 C      1 WT(63),PSYM(7),QSYM(7),MSYM(7),CNSTI(7),AKEEP(7),XKEEP(7,7),
0072 C      2 VAL(63),DRV(63,7)
0073 C
0074 C
0075 C      COMMON /SYSTEM/ SYS(514),NSYS,NSCTR
0076 C
0077 C      COMMON /SSM4N/ A(7),XCNST(7),T(63),NA,ITER,SERSQ,CALM,CORRL(21),
0078 C      1 GUESS(7),CNSTR(7),JORDR(7),AEDIT(2,7)
0079 C
0080 X      WRITE FREE (12) '<15> AT SWHFIT3 INPUT, WTY VALUES ARE:'
0081 X      CALL RLOUT(12,WTY,63)
0082 C
0083 C... SET INITIAL VALUES, LIMITS
0084 C
0085 C
0086 C      IF (NLO.LT.1) NLO=1
0087 C      IF (NUP.GT.63) NUP=63
0088 C
0089 C
0090 C
0091 C      ERLIM=0.001
0092 C      LIMIT=50
0093 C      ITER=0
0094 C      SWHFIT=-3588.
0095 C      IER=-10
0096 C      WTI=0.
0097 C      SQMIN=1.E06
0098 C      SERSQ=SQMIN
0099 C
0100 C..... (USED TO REMOVE BIASES HERE.) NOW CHECK THAT INPUT VALUES ARE
0101 C... WITHIN ALLOWED LIMITS (RETURN WITH NO FURTHER SWH WORK IF
0102 C... NOT WITHIN LIMITS), AND SUM INPUT WEIGHTING FACTORS FOR
0103 C... NORMALIZATION.
0104 C
0105 C      DO 5 I=NLO,NUP
0106 C      WTI=WTI+WTY(I)
0107 C      VI=VIN(I)
0108 C      IF ((VI.LT.-25.).OR.(VI.GT.500.)) RETURN
0109 C      5 CONTINUE
0110 C
0111 C      DO 10 I=NLO,NUP
0112 C      10 WT(I)=WTY(I)/WTI
0113 C
0114 C
0115 C
0116 C... SET INITIAL FIT PARAMETER ESTIMATES. SET ALL TERMS FROM GUESS(7)
0117 C.. WHICH IS ASSUMED TO SET VALUES NOT BEING FITTED AS WELL AS
0118 C.. THE INITIAL GUESSES FOR THOSE BEING FITTED.

```

```

0119 C
0120 DO 9 I=1,7
0121 9 A(I)=GUESS(I)
0122 C
0123 C--- AT THIS POINT IN PROGRAM, SET THE INVERSE-VARIANCE CONSTRAINTS
0124 C-- TO BE ADDED TO THE ON-DIAGONAL TERMS BELOW AT "DO 52" STATEMENT.
0125 C-- PUT TEST TO AVOID AN INPUT STANDARD DEVIATION ESTIMATE LESS THAN
0126 C-- ABOUT 0.033 .
0127 C
0128 IJK=0
0129 DO 12 II=1,NA
0130 I=JCRDR(II)
0131 IF (I.EQ.6) IJK=II
0132 C... CHECKS WHETHER POINTING ANGLE IS ONE OF PARAMETERS BEING FITTED.
0133 C.. SINCE THERE MAY BE CORRECTION SUBSEQUENTLY TO AVOID A NEGATIVE ANGLE.
0134 VI=CNSTR(I)**2
0135 IF (VI.LT.0.001) VI=0.001
0136 CNSTI(I)=1./VI
0137 12 AKEEP(I)=A(I)
0138 C
0139 C
0140 C
0141 C--- FILLV GETS THE 63 SAMPLED VALUES FROM THE FFT-CONVOLUTION
0142 C--- PROGRAM. THESE ARE IN ASCENDING ORDER IN INDEPENDENT VARIABLE.
0143 C
0144 6 CALL FILLV(VAL)
0145 EOLD=0.
0146 IER=1
0147 DO 13 I=NLO,NUP
0148 VI=VIN(I)-VAL(I)
0149 13 EOLD=EOLD + VI*VI*WT(I)
0150 C
0151 C
0152 X WRITE FREE (12) '<15> IN SWHFIT, THE VIN VALUES ARE :'
0153 X CALL RLOUT(12,VIN,63)
0154 X WRITE FREE (12) '<15> AND FIRST-CALL VAL FROM FILLV ARE :'
0155 X CALL RLOUT(12,VAL,63)
0156 C
0157 SGMIN=EOLD
0158 C
0159 C
0160 IF (IXDBG.NE.1) GO TO 11
0161 C
0162 WRITE FREE (12) '<15> SWHFIT; NA =',NA,' ITERATION RESULTS ARE:'
0163 WRITE (12,14) IER,EOLD,A
0164 14 FORMAT(' IER =',I3,', SERSQ =',G14.6,'; A(.) = :',(' ',7F11.5))
0165 C
0166 C... STATEMENT 11 IS RETURN POINT FOR NEXT ITERATION IN FITTING . .
0167 C
0168 11 CONTINUE
0169 C
0170 C... ZERO UPPER PART, SYMMETRIC MATRIX . .
0171 C
0172 DO 20 I=1,NA
0173 DO 15 J=1,I
0174 15 XIAT(J,I)=0.
0175 20 ECOLN(I)=0.
0176 XFRCT=1.
0177 C---PUTTING A COMMENT IN FOLLOWING CARD WILL REMOVE EFFECTS OF
0178 C-- THIS ATTEMPT (2/17/78) AT REDUCING SIZE OF FIRST FEW "STEPS"

```

```

0179 C-- IN PARAMETER SPACE . . .
0180 IF (ITER.LT.4) XFRCT=(1.+FLOAT(ITER))/5.
0181 C
0182 ELIM=ERLIM*XFRCT
0183 SERSQ=0.
0184 C
0185 C--- SUBROUTINE FILLD SETS UP THE (63,7) DERIVATIVE ARRAY BY MAKING
0186 C--- STEPS IN THE VALUES OF THE PARAMETERS A(7); THE STEP SIZES
0187 C--- TAKEN ARE CARRIED BY STPRM(7), AND THE ORDER OF THE DERIVATIVES
0188 C--- IN DRV(63,7) IS SET BY JORDR. FILLV MUST HAVE BEEN CALLED BEFORE
0189 C--- FILLD; THE HELD VALUE IN VAL(63) FROM THE CALL TO FILLV IS USED IN
0190 C--- FILLING DRV(63,7). INCIDENTALLY, NUMERICAL DERIVATIVE IS NOT
0191 C--- DONE IN CASE OF THE AMPLITUDE AND BASELINE, SO THE VALUES SET IN
0192 C--- STPRM(1) AND STPRM(4) ARE IRRELEVANT.
0193 C
0194 C
0195 C
0196 CALL FILLD(VAL,DRV,STPRM)
0197 C
0198 C--- "DO 30" LOOP FILLS UPPER HALF OF THE SYMMETRIC MATRIX, ALSO THE
0199 C--- COLUMN VECTOR . . .
0200 DO 30 JP=NLO,NUP
0201 WT=WT(JP)
0202 DY=YIN(JP)-VAL(JP)
0203 C
0204 C
0205 DO 35 JA=1,NA
0206 PVECT(JA)=DRV(JP,JA)
0207 35 CONTINUE
0208 C
0209 DO 30 I=1,NA
0210 VI=PVECT(I)*WT
0211 DO 40 J=1,I
0212 40 XMAT(J,I)=XMAT(J,I)+VI*PVECT(J)
0213 30 BCOLM(I)=BCOLM(I)+VI*DY
0214 C
0215 C
0216 C--- NOW ADD ON-DIAGONAL CONSTRAINT ELEMENTS TO THE SYMMETRIC MATRIX;
0217 C--- THESE ARE FROM THE A PRIORI INFORMATION ON VARIATION EXPECTED IN
0218 C--- THE PARAMETERS TO BE FITTED (REF. CONVERSATION WITH BILL WELLS OF
0219 C--- WASC1, ABOUT 25/25/78 ). THE CONSTRAINT IS THE INVERSE OF THE INPUT
0220 C--- VARIANCE ESTIMATE, MINIMUM ALLOWED VARIANCE OF 0.001 .
0221 C
0222 C
0223 C
0224 DO 52 I=1,NA
0225 JJ=JORDR(I)
0226 52 XMAT(I,I)=XMAT(I,I)+CNST1(JJ)
0227 C
0228 C--- CALL HERE THE SYMMETRIC MATRIX INVERSION ROUTINE FROM J. MCMILLAN
0229 C
0230 C
0231 CX WRITE FREE (12) 'BEFORE MATRIX INVERSION, INPUT MATRIX : '
0232 CX DO 779 I=1,NA
0233 CX 779 WRITE (12,780) (XMAT(J,I), J=1,I)
0234 CX 780 FORMAT(' ', 8G14.6)
0235 CALL SYMINV(XMAT,NA,IFAIL,7,PSYM,OSYM,MSYM)
0236 CX IF (IFAIL.NE.0) WRITE FREE (12) 'MATRIX INVERSION FAILED . .'
0237 CX WRITE FREE (12) ' AFTER MATRIX INVERSION, OUTPUT MATRIX : '
0238 CX DO 731 I=1,NA

```

```

0239 CX 791 WRITE (12,780) (XMAT(J,I), J=1,I)
0240 C
0241 CX WRITE FREE (12) 'SVMINV IFAIL SHOULD = 0; IT IS ',IFAIL
0242 C
0243 IF (IFAIL.NE.0) GO TO 1000
0244 C
0245 C
0246 DO 66 I=1,NA
0247 II=JORDR(I)
0248 ACLM=0.
0249 J=1,NA
0250 IF (J.LT.I) GO TO 60
0251 ACLM=ACLM+XMAT(I,J)*BCOLM(J)
0252 GO TO 62
0253 60 ACLM=ACLM+XMAT(J,I)*BCOLM(J)
0254 62 CONTINUE
0255 C
0256 IF (II.NE.6) GO TO 64
0257 C
0258 C... THE II=6 PARAMETER IS POINTING ANGLE; FOLLOWING TREATMENT IS AD
0259 C... HOC AND SPECIFIC TO SEASAT-1 CASE (07/25/80).
0260 IF (ABS(A(6)).LT.0.025) ACLM=ACLM/5.
0261 A(6) = A(6) + XFRCT*ACLM
0262 IF ( (A(6).LT.-4.) .OR. (A(6).GT.4.) ) A(6) = A(6)/10.
0263 GO TO 65
0264 C
0265 64 A(II) = A(II) + XFRCT*ACLM
0266 C
0267 C
0268 65 CONTINUE
0269 C
0270 C--- PUT FOLLOWING STEP IN TO AVOID NEGATIVE RISETIME.
0271 C
0272 IF (A(3).LT.1.E-06) A(3) = 1.E-06
0273 C
0274 70 CALL FILLV(VAL)
0275 C--- RECALCULATE VALUES OF THE SAMPLED WAVEFORM FUNCTION FOR THE NEW,
0276 C--- UPDATED ESTIMATES OF THE PARAMETERS A(7).
0277 SERSQ=0.
0278 DO 175 I=NLO,NUP
0279 VI=VIN(I) - VAL(I)
0280 175 SERSQ=SERSQ+VI*VI*WT(I)
0281 ITER=ITER+1
0282 C
0283 C
0284 IF (IXDBG.EQ.1) WRITE (12,14) ITER,SERSQ,A
0285 C
0286 C
0287 C..... CHECK THAT WE KEEP COEFFICIENTS PRODUCING MINIMUM SUM ERRORS**2
0288 IF (SERSQ.GE.SGMIN) GO TO 185
0289 DO 180 I=1,NA
0290 DO 179 J=1,I
0291 179 XKEEP(J,I)=XMAT(J,I)
0292 II=JORDR(I)
0293 180 AKEEP(II)=A(II)
0294 SGMIN=SERSQ
0295 185 CONTINUE
0296 C
0297 C
0298 C

```

```

0299 C.... TEST; AVOID TRYING FOR ABSURDLY SMALL RESIDUALS ABOUT FIT . . .
0300     IF (SERSQ.LT.4.4E-05) GO TO 3000
0301 C
0302     IF (ITER.GE.LIMIT) GO TO 2000
0303     IF (SERSQ.GT.EOLD) GO TO 5000
0304 C
0305     IF ( ((EOLD-SERSQ)/EOLD).LE.ELIM ) GO TO 3000
0306 190   EOLD=SERSQ
0307       GO TO 11
0308 C
0309 C
0310 C--- FOR THE SEASAT WAVEFORM CASE, CHECK IF ERRORS**2 INCREASED; DON'T
0311 C--- MAKE ERROR EXIT IF (FRACTIONAL) INCREASE IS LESS THAN 10 TIMES
0312 C--- LIMIT. ALSO CHECK ABS. MAGNITUDE OF SERSQ; VALUE LESS THAN .0099
0313 C-- IS A INDIVIDUAL S&H GATE STANDARD DEVIATION OF ABOUT 0.015 M,
0314 C-- WHICH WE ASSUME TO BE AN ADEQUATE LOWER LIMIT TO THE SEASAT
0315 C-- SITUATION WHEN USING ONLY LAST 45 GATES.
0316 5000  IF (SERSQ.LE.0.0099) GO TO 3000
0317       IF (ABS((SERSQ-EOLD)/EOLD).LE.10.*ERLIM) GO TO 190
0318 C
0319     IER=-1
0320 C---SUM ERRORS**2 INCREASED
0321 C
0322     GO TO 2500
0323 C
0324 C
0325 1000  IER=-2
0326 C--- MATRIX INVERSION FAILURE (SINGULAR MATRIX . . .)
0327 C
0328     GO TO 2500
0329 2500  IER=-3
0330 C--- ITERATION COUNT EXCEEDED
0331 C
0332 C
0333 C
0334 C... IF ITER .GT. 2, FIGURE THAT SOME SORT OF SOLUTION EXISTS, SO SET
0335 C.. IER=1 AND RETRIEVE THE MINIMUM-VALUE-PRODUCING SET OF A(.) AND THE
0336 C.. RESULTING XMAT(...) VALUES IF NECESSARY . .
0337 2500  IF (ITER.LE.2) GO TO 4500
0338       IER=1
0339       IF (SERSQ.LT.SQMIN) GO TO 3000
0340       DO 2510 I=1,NA
0341         DO 2505 J=1,I
0342 2505  XMAT(J,I)=XKEEP(J,I)
0343       II=JORDR(I)
0344 2510  A(II)=AKEEP(II)
0345       SERSQ=SQMIN
0346 C
0347 C
0348 C
0349 3000  CONTINUE
0350 C
0351 C--- USE VALUES FROM XMAT (AT LAST ITERATION & AFTER INVERSION) TO
0352 C-- FIND CORRELATIONS WHICH WILL THEN BE SET INTO ARRAY CORRL(21) IN
0353 C-- ORDER : 2,1 3,1 3,2 4,1 4,2 4,3, ETC. . NOTE THAT FIRST THE
0354 C-- SQUARE ROOTS OF DIAGONAL ELEMENTS WILL BE TAKEN, FOR CONVENIENCE.
0355 C-- ALSO NOTE THAT ORDER IN THIS CORRELATION ARRAY IS IN TERMS OF
0356 C-- THE ORDER IN WHICH THE PARAMETERS WERE FITTED, NOT THE ORDER IN
0357 C-- WHICH THEY ARE IN A(.) .
0358 C

```



```

0359      DO 3001 I=1,NA
0360 3001  CORR(I)=0.
0361 C
0362      IJ=0
0363      DO 3005 J=2,NA
0364      JM=J-1
0365      DO 3005 I=1,JM
0366      IJ=IJ+1
0367      CORR(IJ)=XMAT(I,J)/(XMAT(I,I)*XMAT(J,J))
0368 3005  CONTINUE
0369 C
0370 C
0371 C--- 3010 REACHED WHEN LINEFIT CONVERGED, PRODUCED PARAMETER ESTIMATES.
0372 3010  CONTINUE
0373 C
0374 C... CHECK LINEFIT PARAMETERS AGAINST EDIT LIMITS, SIGNAL BY IER.GT.1
0375 C
0376      DO 3012 II=1,NA
0377      I=JORDR(II)
0378      VI=A(I)
0379 3012  IF ( (VI.LT.AEDIT(1,I)).OR.(VI.GT.AEDIT(2,I)) ) GO TO 4000
0380      GO TO 4500
0381 C
0382 4000  IER=1+I
0383 C
0384 C--- COMPUTE SWH ESTIMATE, WITH POSSIBILITY OF NEGATIVE SWH . . .
0385 4500  DIFQ=A(3)**2-CALM*CALM
0386      SWHFIT=0.6*SIGN(SQRT(ABS(DIFQ)),DIFQ)
0387 C
0388      RETURN
0389      END

```

```

0001      SUBROUTINE SYMINV (A,N,IFAIL,NROW,P,Q,M)
0002 C...PROGRAMMED BY: JIM McMILLAN - WASC - REVISED 03/08/78
0003 C...PURPOSE:      TO COMPUTE THE INVERSE OF A SYMMETRIC MATRIX
0004      DIMENSION A(NROW,1),P(1),Q(1),M(1)
0005      IFAIL = 0
0006      DO 1 I=1,N
0007      1 M(I) = 1
0008 C...BEGIN CALCULATION.
0009      DO 14 I = 1,N
0010 C...SEARCH FOR PIVOT.
0011      BIG = 0.0
0012      DO 4 J=1,N
0013      TEST = ABS(A(J,J))
0014      IF (TEST-BIG) 4,4,2
0015      2 IF (M(J)) 15,4,3
0016      3 BIG = TEST
0017      K = J
0018      4 CONTINUE
0019 C...PREPARATION FOR ELIMINATION STEP NO. 1.
0020      M(K) = 0
0021      Q(K) = 1.0 / A(K,K)
0022      P(K) = 1.0
0023      A(K,K) = 0.0
0024      KP1 = K + 1
0025      KM1 = K - 1
0026      IF (KM1) 15,8,5
0027      5 DO 7 J=1,KM1
0028      P(J) = A(J,K)
0029      Q(J) = A(J,K) * Q(K)
0030      IF (M(J)) 15,7,6
0031      6 Q(J) = -Q(J)
0032      7 A(J,K) = 0.0
0033      8 IF (K-N) 9,13,15
0034      9 DO 12 J=KP1,N
0035      P(J) = A(K,J)
0036      IF (M(J)) 15,10,11
0037      10 P(J) = -P(J)
0038      11 Q(J) = -A(K,J) * Q(K)
0039      12 A(K,J) = 0.0
0040 C...ELIMINATION PROPER.
0041      13 DO 14 J=1,N
0042      DO 14 K=J,N
0043      14 A(J,K) = A(J,K) + P(J)*Q(K)
0044      RETURN
0045 C...ERROR EXIT.
0046      15 IFAIL = 1
0047      RETURN
0048      END

```

APPENDIX C.
SOURCE LISTINGS FOR SUBROUTINES FILLV AND FILLD

```

0001 C      DP0:GHAYNE:FILLV3.FR      REVISED 10/15/80, APPROX. 1320 HOURS
0002 C
0003 C      G.S.HAYNE          WFC/DAS          09/17/80
0004 C
0005 C
0006 C
0007 C      THIS VERSION USES THE RADIX 8-4-2 FFT ROUTINES; THE VERSION
0008 C      IN FILE FILLV2.FR USES THE RADIX 4-2 ROUTINES. FILLV3 USES
0009 C      THE VALUES OF SYS(514) WHICH ARE PASSED BY A NEW LABELLED
0010 C      COMMON ARRAY /SYSTEM/
0011 C
0012 C      SUBROUTINE EVALUATES SEASAT-1 WAVEFORMS USING FFT TECHNIQUES TO
0013 C      PERFORM CONVOLUTION OF : 1) SYSTEM POINT-TARGET RESPONSE,
0014 C      2) SEA-SURFACE ELEVATION DISTRIBUTION, AND 3) FLAT-SEA RESPONSE.
0015 C      THIS ROUTINE IS SET UP FOR A 512-POINT TRANSFORM, AND USES FFT
0016 C      SUBROUTINES FFA & FFS TOGETHER WITH REQUIRED SUBROUTINES FOR
0017 C      FFA AND FFS (THESE ARE R2TR,R4TR,R8TR,R4SYN,R8SYN,
0018 C      ORD1, AND ORD2).
0019 C      ALSO USES SUBROUTINES GTSYS, GTSEA2, AND GTFSR2 (AND CPOUT &
0020 C      RLOUT ON A /X COMPILE SWITCH).
0021 C
0022 C      SUBROUTINE FILLV(VAL)
0023 C      COMPILER STATIC
0024 C
0025 C      INTEGER I1ST/0/,NNP/512/,NP2/514/,NC2/257/
0026 C
0027 X      INTEGER JDATE(3),JTIME(3)
0028 C
0029 C
0030 C      REAL VAL(63),SEA(514),FSR(514),RES(514)
0031 C      COMPLEX CTSYS(257),CTSEA(257),CTFSR(257),CTRES(257),CPHAS
0032 C      EQUIVALENCE (SYS(1),CTSYS(1)),(SEA(1),CTSEA(1)),(FSR(1),CTFSR(1)),
0033 C      1 (RES(1),CTRES(1)),(A(1),AMPLI),(A(2),TIM0),(A(3),SIGMA),
0034 C      2 (A(4),BSLIN),(A(5),XLMDA),(A(6),XIDEG),(A(7),XKURT)
0035 C
0036 C      COMMON /SSM4N/ A(7),XCNST(7),T(63),NA,ITER,SERSQ,CALM,CORRL(21),
0037 C      1 GUESS(7),CNSTR(7),JORDR(7),AEDIT(2,7)
0038 C
0039 C      COMMON /SYSTEM/ SYS(514),NSYS,NSCTR
0040 C
0041 C      C... NSYS=# VALUES AT THE POINT AFTER WHICH THE REST OF THE 514
0042 C      C... SYS(.) VALUES EQUAL ZERO, AND NSCTR=INDEX OF PEAK OF THE
0043 C      C... POINT-TARGET RESPONSE.
0044 C
0045 C      W A R N I N G I I I
0046 C      C... SYS(.) CONTAINS THE TRANSFORM AFTER THE FIRST CALL TO FILLV3, SO
0047 C      C... DO NOT CHANGE SYS(.) IN THE EXTERNAL PROGRAM.
0048 C      C... ALSO, IF THE INFORMATION IS TO BE PRINTED OUT, DO THIS BEFORE
0049 C      C... THE FIRST FILLV3 CALL.
0050 C
0051 C
0052 C      C---- NNP IS # POINTS IN TRANSFORM (MUST BE POWER OF 2), NP2 IS 2 MORE
0053 C      C      THAN NNP, AND NC2 IS # COMPLEX TRANSFORM VALUES WHICH WILL RESULT
0054 C      C      FROM THIS.
0055 C
0056 X      CALL DATE(JDATE,I)
0057 X      JDATE(3) = JDATE(3) - 1900
0058 X      CALL TIME(JTIME,I)

```

```

0059 X      WRITE (12,133) JTIME,JDATE
0060 X 133   FORMAT(' FILLV3 DEBUG AT ',I2,2(':',I2),' ON ',I2,2('/',I2))
0061 C
0062       IF (I1ST.NE.0) GO TO 250
0063       I1ST=1
0064       SMSYS=0.
0065       DO 150 I=1,NSYS
0066   150   SMSYS = SMSYS + SYS(I)
0067       J = NSYS + 1
0068       DO 155 I=J,514
0069   155   SYS(I) = 0.
0070 C
0071 C..... IF DESIRED, CAN REPLACE THE BUILT-IN SAMPLED SYSTEM RESPONSE
0072 C      FUNCTION HERE BY ANOTHER ROUTINE GTSYS BY EXECUTING
0073 C      THE FOLLOWING STATEMENT (I.E., REMOVE "C" IN COL.1).
0074 C      CALL GTSYS(NNP,SYN,SMSYS)
0075 C
0076 C--- NOW DO THE TRANSFORM OF SYSTEM IMPULSE RESPONSE
0077 X      WRITE FREE (12) '<15> INPUT REAL SYS(.) : '
0078 X      WRITE FREE (12) ' SMSYS = ',SMSYS
0079 X      CALL RLOUT(12,SYN,NNP)
0080       CALL FFA(SYN,NNP)
0081 X      WRITE FREE (12) '<15> COMPLEX, TRANSFORMED SYS(.) : '
0082 X      CALL CPOUT(12,SYN,NC2)
0083 C
0084 C-----NOW SET UP SEA SURFACE ELEVATION DISTRIBUTION.
0085   250 CALL GTSEA(NNP,SEA,SMSEA)
0086 C--- NOW DO TRANSFORM OF SEA SURFACE ELEVATION DISTRIBUTION.
0087 X      WRITE FREE (12) '<15> INPUT REAL SEA-SURFACE SEA(.) : '
0088 X      WRITE FREE (12) ' SMSEA = ',SMSEA
0089 X      CALL RLOUT(12,SEA,NNP)
0090       CALL FFA(SEA,NNP)
0091 X      WRITE FREE (12) '<15> COMPLEX, TRANSFORMED SEA(.) : '
0092 X      CALL CPOUT(12,SEA,NC2)
0093 C
0094 C----- SET UP THE FLAT-SEA RESPONSE FUNCTION.
0095       CALL GTFSR(NNP,FSR,SMFSR)
0096 C--- TRANSFORM THE FLAT-SEA RESPONSE FUNCTION
0097 X      WRITE FREE (12) '<15> FLAT-SEA IMPULSE RESPONSE FSR(.) : '
0098 X      WRITE FREE (12) ' SMFSR = ',SMFSR
0099 X      CALL RLOUT(12,FSR,NNP)
0100       CALL FFA(FSR,NNP)
0101 X      WRITE FREE (12) '<15> COMPLEX, TRANSFORMED FSR(.) : '
0102 X      CALL CPOUT(12,FSR,NC2)
0103 C
0104 C----- FORM AMPLITUDE NORMALIZATION ANORM, THEN SET UP PHASE MULTI-
0105 C      PLIER DELTA FACTOR DPHI (XNCTR IS FOR POSSIBLE LATER SHIFT OF
0106 C      ENTIRE RESULT IF DESIRED. AS NOW SET, FIRST 119 VALUES OF RES(.)
0107 C      ARE THOSE SPANNING THE DESIRED 63 SAMPLER SET OF SEASAT-1 AND
0108 C      THE CENTER IS AT THE 60TH GATE IF XNCTR IS 0 .)
0109       XNCTR = 0.
0110       PHI = 0.
0111       DPHI = -( (XNCTR-24.5-FLOAT(NSCTR)) + TIM3/1.5625)*
0112   1      6.2831853/FLOAT(NNP)
0113       ANORM = 1./(SMSYS*SMSEA)
0114       CTRES(1) = CMPLX(ANORM,0.)*CTSYS(1)*CTSEA(1)*CTFSR(1)
0115       DO 500 I=2,NC2
0116         PHI = PHI + DPHI
0117         CPHAS = CMPLX(ANORM*COS(PHI),ANORM*SIN(PHI))
0118   500   CTRES(I) = CPHAS*CTSYS(I)*CTSEA(I)*CTFSR(I)

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

0119 C
0120 C----- ABOVE WAS A COMPLEX MULTIPLICATION (IN TRANSFORM DOMAIN)
0121 C
0122 C--- DO THE INVERSE TRANSFORM TO GET FINAL CONVOLUTION RESULT
0123 C
0124 X      WRITE FREE (12) '<15> COMPLEX, TRANSFORM RESULT RES(.):'
0125 X      WRITE FREE (12) ' ANORM = ',ANORM
0126 X      CALL CPOUT(12,RES,NC2)
0127      CALL FFS(RES,NNP)
0128 X      WRITE FREE (12) '<15> FINAL REAL RESULT RES(.):'
0129 X      CALL RLOUT(12,RES,NNP)
0130 C
0131      IF (ABS(BSLIN).LE.(AMPLI*0.1E-05)) GO TO 550
0132      DO 530 I=1,NNP
0133      530 RES(I) = RES(I) + BSLIN
0134      550 CONTINUE
0135 C
0136 C
0137 C      TRANSFER DATA TO FINAL OUTPUT ARRAY VAL(63) AND WRITE OUT.
0138 C
0139      DO 610 I=1,29
0140      J=I+I
0141      VAL(I+34)=RES(J+61)
0142      610 VAL(I)=RES(J-1)
0143 C
0144      DO 620 I=30,34
0145      620 VAL(I)=RES(I+28)
0146 C
0147      J=63
0148 C
0149 C--- NOW RETURN TO CALLING PROGRAM
0150 C
0151      RETURN
0152      END

```

```

0001 C      DP0:GHAYNE:FILLD3.FR      REVISED 09/25/80, APPROX. 0930 HOURS
0002 C
0003 C      G.S.HAYNE      WFC/DAS      09/25/80
0004 C
0005 C      FILLD3 IS EXACTLY SAME AS FILLD2 EXCEPT FOR THE ADDITION
0006 C      OF ANOTHER LABELLED COMMON AREA /SYSTEM/. THIS COMMON IS
0007 C      FOR COMMUNICATION OF INITIAL SYSTEM POINT-TARGET RESPONSE
0008 C      FUNCTION VALUES BETWEEN FILLV (FILE FILLV3) AND THE MAIN,
0009 C      CALLING PROGRAM; THIS WAS NECESSARY FOR SSFIT3 AND SIMILAR
0010 C      PROGRAMS...      GSH      09/25/80
0011 C
0012 C
0013 C
0014 C      THIS ROUTINE SAME AS FILLD.FR EXCEPT FOR DIFFERENT COMMON
0015 C      AREA, SSM4N.
0016 C
0017 C      A SUBROUTINE TO FILL AN ARRAY DRV(63,7) CONTAINING 63 SAMPLE
0018 C      VALUES OF THE UP TO 7 DERIVATIVES FOR SEASAT WAVEFORM CASE;
0019 C      THE ORDER OF THE DERIVATIVE TERMS IN DRV IS SET BY JORDR(7).
0020 C      FILLD IS A COMPANION TO FILLV; IT REQUIRES THAT FILLV HAS BEEN
0021 C      CALLED ALREADY. SEE FURTHER COMMENTS IN THE FITTING ROUTINE
0022 C      SWHFIT WHICH CALLS FILLD AND FILLV. THESE ROUTINES ARE PART
0023 C      OF THE NEW PROGRAMS USING THE FFT-CONVOLUTION TECHNIQUE WITHIN
0024 C      WAVEFORM GENERATION. DEBUG PRINTOUT TO (ASSUMED ALREADY OPENED)
0025 C      DEVICE 12 IS PROVIDED ON X COMPILE SWITCH.
0026 C
0027 C
0028 C      FILLD2 REQUIRES SUBROUTINES FILLV,GTSYS,GTSEA2,GTFSR2,FR2TR,FR4T
0029 C      FR4SYN,FORD1, AND FORD2.
0030 C
0031 C      THE DERIVATIVES ARE NUMERICALLY FOUND BY STEPPING A(7), THE
0032 C      PARAMETER VALUES BY AN AMOUNT SET BY STPRM (EXCEPT THAT FIRST
0033 C      AND FOURTH TERMS, AMPLITUDE & BASELINE, ARE FOUND DIFFERENTLY).
0034 C
0035 C      SUBROUTINE FILLD(VAL,DRV,STPRM)
0036 C      COMPILER STATIC
0037 C
0038 C      COMMON /SYSTEM/ SYS(514),NSYS,NSCTR
0039 C
0040 C      COMMON /SSM4N/ A(7),XCNST(7),T(63),NA,ITER,SERSQ,CALM,CORRL(21),
0041 C      1 GUESS(7),CNSTR(7),JORDR(7),AEDIT(2,7)
0042 C
0043 C      EQUIVALENCE (AMPL1,A(1)),(BSLIN,A(4))
0044 C
0045 C
0046 C      REAL VAL(63),DRV(63,7),STPRM(7)
0047 X      REAL XTMP(63)
0048 C
0049 C
0050 X      WRITE FREE (12) ' DEBUG IN FILLD; CALL FILLV & PRINT RESULTS : '
0051 X      CALL FILLV(VAL)
0052 X      CALL RLOUT(12,VAL,63)
0053 C
0054 C
0055 C--- LOOP TO 400 FOR THE NA DERIVATIVES NEEDED.
0056 C
0057 C      DO 400 K=1,NA
0058 C      J=JORDR(K)

```

```

0039 C
0060 C--- DO NUMERICAL STEP PROCEDURE AT 300 FOR EVERYTHING EXCEPT AMPLI AND
0061 C--- BSLIN; FOR THESE WE KNOW EASIER WAY TO GET ANSWER.
0062 C
0063 X WRITE FREE (12) ' FILLO DEBUG: AT COMPUTED GO TO, J & K = ',J,K
0064 C
0065 GO TO (100,300,300,200,300,300,300), J
0066 C
0067 C--- 100 IS STATEMENT REACHED WHEN AMPLI DERIV IS NEEDED.
0068 100 CONTINUE
0069 X WRITE FREE (12) ' FILLO DEBUG: AT STATEMENT # 100'
0070 TMP=AMPLI
0071 IF (TMP.LT.1.E-05) TMP=1.E-05
0072 DO 110 I=1,63
0073 110 DRV(I,K)=(VAL(I)-BSLIN)/TMP
0074 GO TO 400
0075 C
0076 C--- 200 IS STATEMENT REACHED WHEN BSLIN DERIV IS NEEDED.
0077 200 CONTINUE
0078 X WRITE FREE(12) ' FILLO DEBUG: AT STATEMENT # 200'
0079 DO 210 I=1,63
0080 210 DRV(I,K)=1.0
0081 GO TO 400
0082 C
0083 C--- 300 IS STATEMENT FOR NUMERICAL STEP ESTIMATION OF THE DERIVATIVE.
0084 300 CONTINUE
0085 X WRITE FREE (12) ' FILLO DEBUG: AT STATEMENT # 300'
0086 ATMP=A(J)
0087 STEP=STPRM(J)
0088 A(J)=A(J)+STEP
0089 C--- ABOVE STATEMENTS SWAPPED IN NEW VALUE FOR THE DESIRED A(.)
0090 C--- NOW STORE ELEMENTS OF VAL(.) TEMPORARILY IN DRV(.,J)
0091 X WRITE FREE (12) ' FILLO DEUG: J & A(J) = ',J,A(J)
0092 DO 310 I=1,63
0093 310 DRV(I,K)=VAL(I)
0094 C
0095 C--- GET NEW WAVEFORM WITH NEW (TEMPORARY, STEPPED) PARAMETER VALUES.
0096 C
0097 X WRITE FREE (12) '<LF> TEMPORARY A(.) :',(A(I), I=1,7)
0098 C
0099 CALL FILLV(VAL)
0100 X WRITE FREE (12) ' AND FOR THESE VALUES, FUNCTION IS :'
0101 X CALL RLOUT(12,VAL,63)
0102 C
0103 C
0104 C--- NOW FORM DERIV. ESTIMATES & SWAP VALUES BACK TO ORIGINAL..
0105 C
0106 DO 320 I=1,63
0107 TMP=DRV(I,K)
0108 DRV(I,K)=(VAL(I)-TMP)/STEP
0109 X XTMP(I)=DRV(I,K)
0110 320 VAL(I)=TMP
0111 C
0112 A(J)=ATMP
0113 C
0114 C
0115 X WRITE FREE (12) ' <LF> STEP-DERIV. VALUES ARE :'
0116 X CALL RLOUT(12,XTMP,63)
0117 C
0118 C

```

```

0119 X      WRITE FREE (12) ' RESTORED FUNCTION VALUES : '
0120 X      CALL RLOUT(12,VAL,63)
0121 C
0122 C
0123 C--- BOTTOM OF MAIN DERIVATIVE LOOP
0124 C
0125 400    CONTINUE
0126 C
0127      RETURN
0128      END

```


APPENDIX D.

SOURCE LISTINGS FOR SUBROUTINES GTFSK AND GTSEA

```

0001 C      DP0:GHAYNE:GTFSR2.FR      REVISED 11/13/80, 1505 HRS
0002 C
0003 C      G.S.HAYNE      APPL.SCI.ASSOC'S      07/16/80
0004 C
0005 C      THIS VERSION ADMITS A NEGATIVE ANGLE, BUT FAKES THE RESULT
0006 C      TO GIVE A FASTER-THAN-ZERO-DEGREES PLATEAU DECAY. 07/16/80
0007 C
0008 C----- SUBROUTINE GTFSR2 FILLS THE FLAT-SURFACE RESPONSE, USING I0
0009 C      TERM ONLY FROM EXPANSION IN GARY BROWN'S PAPER. A POWER SERIES
0010 C      FROM ABRAMOWITZ & STEGUN IS USED TO EVALUATE I0. THIS VERSION OF
0011 C      GTFSR USES 230 NON-ZERO VALUES OF THE FLAT-SURFACE RESPONSE, AND
0012 C      ASSUMES THAT NNP > 230. GTFSR2 IS DERIVED FROM GTFSR BUT IS
0013 C      FOR THE NEW FFT-BASED CONVOLUTION USING ONLY 512-POINT
0014 C      FFT.
0015 C
0016 C      SUBROUTINE GTFSR(NNP,FSR,SMFSR)
0017 C      REAL FSR(1)
0018 C      COMMON /SSM4N/ AMPLI,TIM0,SIGMA,BSLIN,XLMDA,XIDEG,XKURT
0019 C      NFSR = 230
0020 C      SMFSR = 0.
0021 C      DT = 1.5625
0022 C      T = -DT/2.
0023 C
0024 C... TEST FOR (IMPOSSIBLE) NEGATIVE ANGLE; IF PRESENT, CHOOSE BRANCH WHICH
0025 C... EFFECTIVELY INCREASES THE DLTA AT < ZERO DEGREES POINTING (CHANGES
0026 C... MADE ON 07/16/80)
0027 C
0028 C      IF (XIDEG.GT.0.) GO TO 20
0029 C
0030 C... THE FOLLOWING STATEMENT CAUSES DLTA TO INCREASE BY A FACTOR OF
0031 C... TWO FOR 1 DEGREE (FICTICIOUS) NEGATIVE SEASAT ANGLE...
0032 C      DLTA = 2.56496E-03*(1.-XIDEG)
0033 C      DO 15 J=1,NFSR
0034 C          T = T + DT
0035 C          Z = AMPLI*EXP(-DLTA*T)
0036 C          FSR(J) = Z
0037 15      SMFSR = SMFSR + Z
0038 C      GO TO 35
0039 C
0040 20      X2RAD = XIDEG/28.64789
0041 C      BETA = 4.35331*SIN(X2RAD)
0042 C      DLTA = 2.554.6E-03*COS(X2RAD)
0043 C      DO 30 J=1,NFSR
0044 C          T = T + DT
0045 C          Z = BETA*SQRT(T)
0046 C----- SELECT WHICH OF THE TWO SERIES TO USE FOR I0(Z)
0047 C      IF (Z.GT.3.75) GO TO 23
0048 C      Z = Z*Z/14.0625
0049 C----- 3.75**2 = 14.0625
0050 C      A = 1. + Z*(3.515623+Z*(3.089942+Z*(1.206749
0051 1      +Z*(0.2659732+Z*(0.0360769+Z*0.0045813))))))
0052 C      GO TO 27
0053 23      A = EXP(Z)/SQRT(Z)
0054 C      Z = 3.75/Z
0055 C      A = A*(0.3989423-Z*(0.03988024+Z*(0.00362018
0056 1      -Z*(0.00163801-Z*(0.01031555-Z*(0.02282967-Z*(0.02895912
0057 2      -Z*(0.01787654-Z*0.00420059)))))))))
0058 27      Z = AMPLI*EXP(-DLTA*T)*A

```

```

0059      FSR(J) = Z
0060      30 SMFSR = SMFSR + Z
0061 C
0062 C
0063      35      CONTINUE
0064 C
0065 C----- FILL REST OF THE ARRAY WITH ZEROES
0066      K = NFSR + 1
0067      NP2 = NNP + 2
0068      DO 40 J = K, NP2
0069      40 FSR(J) = 0.
0070      IIST = 1
0071      RETURN
0072 C
0073      END

```

```

0001 C      DP0:GHAYNE:GTSEA2.FR      REVISED 10/15/80, 1400 HRS
0002 C
0003 C      G.S.HAYNE      APPL.SCI.ASSOC'S      06/20/80
0004 C
0005 C      DISCOVERED IMPORTANT SIGN ERROR IN THIS ROUTINE ON 10/15/80;
0006 C      CHANGED IT AT ABOUT 1400 HOURS. PRIOR TO THIS, ALL
0007 C      GTSEA2 RESULTS WERE RETURNING OPPOSITE SIGN OF SKEWNESS TO
0008 C      THAT INTENDED...
0009 C
0010 C      SUBROUTINE GTSEA2 FILLS ARRAY WITH A SKEWED GAUSSIAN SURFACE
0011 C      ELEVATION DISTRIBUTION, CENTERED ON THE SAMPLE NUMBER 86.
0012 C      ZEROES ENTERED IN ALL OTHER ELEMENTS THAN IN INTERVAL 1 - 171.
0013 C      ASSUMES NNP > 171 . THIS ROUTINE SAME AS GTSEA.FR EXCEPT FOR DIFFERENT
0014 C      COMMON NAME AND FOR THE DISTRIBUTION CENTERED AT 171. IT IS INTENDED
0015 C      FOR USE IN THE 512-POINT FFT PROCESSES.
0016 C
0017 C      SUBROUTINE GTSEA(NNP,SEA,SMSEA)
0018 C      REAL SEA(2)
0019 C      COMMON /SSM4N/ AMPLI,TIM0,SIGMA,BSLIN,XLMCA,XIDEG,XKURT
0020 C----- CONVERT SEA SIGMA TO # GATE INTERVALS
0021 C      XNGTS = SIGMA/1.5625
0022 C----- ZERO WIDTH IS NOT ALLOWED
0023 C      IF (XNGTS.LT.0.001) XNGTS = 0.001
0024 C      WGTS = 0.
0025 C----- ESTABLISH CENTER AT 86TH GATE
0026 C      NCTR = 86
0027 C      SEA(NCTR) = 1.
0028 C      SMSEA = 1.
0029 C      K = NCTR - 1
0030 C      X6 = XLMCA/6.
0031 C      XNDX = 0.
0032 C----- FILL NON-ZERO ELEMENTS OF THE ARRAY
0033 C      DO 20 J=1,K
0034 C      XNDX = XNDX + 1.
0035 C      WGTS = XNDX/XNGTS
0036 C      Z = -WGTS*WGTS/2.
0037 C      IF (Z.LT.-80.) GO TO 10
0038 C      A = EXP(Z)
0039 C      Z = X6*WGTS*(WGTS*WGTS-3.)
0040 C
0041 C
0042 C... REVERSED +,- SIGNS IN A1,A2 BELOW AFTER FINDING SIGN ERROR ON
0043 C... 10/15/80
0044 C
0045 C      A1 = A*(1. - Z)
0046 C      A2 = A*(1. + Z)
0047 C
0048 C
0049 C
0050 C      GO TO 15
0051 C      10 A1 = 0.
0052 C      A2 = 0.
0053 C      15 SEA(NCTR-J) = A1
0054 C      SEA(NCTR+J) = A2
0055 C      20 SMSEA = SMSEA + A1 + A2
0056 C----- NOW FILL ZERO ELEMENTS OF THE ARRAY
0057 C      K = 2*NCTR
0058 C      NP2 = NNP + 2
0059 C
0060 C      DO 30 J=K,NP2
0061 C      30 SEA(J) = 0.
0062 C      RETURN
0063 C      END

```

APPENDIX E.

SOURCE LISTINGS FOR SUBROUTINES FFA AND FFS

```

0001 C      DPO:GHAYNE:FFA.FR      REVISED 07/03/80, APPROX. 1055 HOURS
0002 C
0003 C      G.S.HAYNE      APPL.SCI.ASSOC'S.      07/01/80
0004 C
0005 C      THE FOLLOWING IS ONE OF THE SET OF ROUTINES FOR FAST FOURIER
0006 C      TRANSFORM OF REAL DATA SEQUENCE AS DESCRIBED IN SECTION 1.2 OF
0007 C      "PROGRAMS FOR DIGITAL SIGNAL PROCESSING," ED. BY DIGITAL SIGNAL
0008 C      PROCESSING COMMITTEE OF THE IEEE ASSP, PUBLISHED BY IEEE PRESS,
0009 C      NY,1979. THESE ROUTINES ARE COLLECTIVELY THE "FFA-FFS PACKAGE"
0010 C      WHICH INCLUDES: FFA, FFS, R2TR, R4TR, R8TR, R4SYN, R8SYN, ORD1,
0011 C      AND ORD2.
0012 C
0013 C-----
0014 C      SUBROUTINE:  FFA
0015 C      FAST FOURIER ANALYSIS SUBROUTINE
0016 C-----
0017 C
0018 C      SUBROUTINE FFA(B, NFFT)
0019 C
0020 C      THIS SUBROUTINE REPLACES THE REAL VECTOR B(K), (K=1,2,...,N),
0021 C      WITH ITS FINITE DISCRETE FOURIER TRANSFORM. THE DC TERM IS
0022 C      RETURNED IN LOCATION B(1) WITH B(2) SET TO 0. THEREAFTER, THE
0023 C      JTH HARMONIC IS RETURNED AS A COMPLEX NUMBER STORED AS
0024 C      B(2*J+1) + I B(2*J+2). NOTE THAT THE N/2 HARMONIC IS RETURNED
0025 C      IN B(N+1) WITH B(N+2) SET TO 0. HENCE, B MUST BE DIMENSIONED
0026 C      TO SIZE N+2.
0027 C      SUBROUTINE IS CALLED AS FFA (B,N) WHERE N=2**M AND B IS AN
0028 C      N TERM REAL ARRAY. A REAL-VALUED, RADIX 8 ALGORITHM IS USED
0029 C      WITH IN-PLACE REORDERING AND THE TRIG FUNCTIONS ARE COMPUTED AS
0030 C      NEEDED
0031 C
0032 C      DIMENSION B(2)
0033 C      COMMON /CON/ PII, P7, P7TWO, C22, S22, PI2
0034 C
0035 C      IW IS A MACHINE-DEPENDENT WRITE-DEVICE NUMBER
0036 C
0037 C      IW=12
0038 C
0039 C      PII = 4.*ATAN(1.)
0040 C      PI8 = PII/8.
0041 C      P7 = 1./SQRT(2.)
0042 C      P7TWO = 2.*P7
0043 C      C22 = COS(PI8)
0044 C      S22 = SIN(PI8)
0045 C      PI2 = 2.*PII
0046 C      N = 1
0047 C      DO 10 I=1,15
0048 C          M = I
0049 C          N = N*2
0050 C          IF (N.EQ.NFFT) GO TO 20
0051 10  CONTINUE
0052 C      WRITE (IW,9999)
0053 9999 FORMAT(' NFFT NOT A POWER OF 2 FOR FFA')
0054 C      STOP
0055 20  CONTINUE
0056 C      NSPOW = M/3
0057 C
0058 C      DO A RADX 2 OR RADIX 4 ITERATION FIRST IF CHI IS REQUIRED

```

```

0050 C      IF (M-N8POW*3-1) 50, 40, 30
0051 30      NN = 4
0052          INT = N/NN
0053          CALL R4TR(INT, B(1), B(INT+1), B(2*INT+1), B(3*INT+1))
0054          GO TO 60
0055 40      NN = 2
0056          INT = N/NN
0057          CALL R2TR(INT, B(1), B(INT+1))
0058          GO TO 60
0059 50      NN = 1
0060 C
0071 C      PERFORM RADIX 8 ITERATIONS
0072 C
0073 60      IF (N8POW) 90, 90, 70
0074 70      DO 80 IT=1,N8POW
0075          NN = NN*8
0076          INT = N/NN
0077          CALL R8TR(INT, NN, B(1), B(INT+1), B(2*INT+1), B(3*INT+1),
0078              1      B(4*INT+1), B(5*INT+1), B(6*INT+1), B(7*INT+1), B(1),
0079              2      B(INT+1), B(2*INT+1), B(3*INT+1), B(4*INT+1), B(5*INT+1),
0080              3      B(6*INT+1), B(7*INT+1))
0081 80      CONTINUE
0082 C
0083 C      PERFORM IN-PLACE REORDERING
0084 C
0085 90      CALL ORD1(M, B)
0086          CALL ORD2(M, B)
0087          T = B(2)
0088          B(2) = 0.
0089          B(NFFT+1) = T
0090          B(NFFT+2) = 0.
0091          DO 100 I=4,NFFT,2
0092              B(I) = -B(I)
0093 100      CONTINUE
0094          RETURN
0095          END

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

0001 C      DP0:GHAYNE:FFS.FR      REVISED 07/03/80, APPROX. 1130 HOURS
0002 C
0003 C      G.S.HAYNE      APPL.SCI.ASSOC'S.      07/02/80
0004 C
0005 C      THE FOLLOWING IS ONE OF THE SET OF ROUTINES FOR FAST FOURIER
0006 C      TRANSFORM OF REAL DATA SEQUENCE AS DESCRIBED IN SECTION 1.2 OF
0007 C      "PROGRAMS FOR DIGITAL SIGNAL PROCESSING," ED. BY DIGITAL SIGNAL
0008 C      PROCESSING COMMITTEE OF THE IEEE ASSP, PUBLISHED BY IEEE PRESS,
0009 C      NY, 1979. THESE ROUTINES COLLECTIVELY ARE "FFA-FFS PACKAGE"
0010 C      WHICH INCLUDES: FFA, FFS, R2TR, R4TR, R8TR, R4SYN, R8SYN, ORD1,
0011 C      AND ORD2.
0012 C
0013 C
0014 C -----
0015 C      SUBROUTINE: FFS
0016 C      FAST FOURIER SYNTHESIS SUBROUTINE
0017 C      RADIX 8-4-2
0018 C -----
0019 C
0020 C      SUBROUTINE FFS(S, NFFT)
0021 C
0022 C      THIS SUBROUTINE SYNTHESIZES THE REAL VECTOR B(K), WHERE
0023 C      K=1,2,...,N. THE INITIAL FOURIER COEFFICIENTS ARE PLACED IN
0024 C      THE B ARRAY OF SIZE N+2. THE DC TERM IS IN B(1) WITH
0025 C      B(2) EQUAL TO 0.
0026 C      THE JTH HARMONIC IS STORED AS B(2*J+1) + I B(2*J+2).
0027 C      THE N/2 HARMONIC IS IN B(N+1) WITH B(N+2) EQUAL TO 0.
0028 C      THE SUBROUTINE IS CALLED AS FFS(B,N) WHERE N=2**M AND
0029 C      B IS THE N TERM REAL ARRAY DISCUSSED ABOVE.
0030 C
0031 C      DIMENSION B(2)
0032 C      COMMON /CON1/ PII, P7, P7TWO, C22, S22, PI2
0033 C
0034 C      IW IS A MACHINE-DEPENDENT WRITE-DEVICE NUMBER
0035 C
0036 C      12 = 12
0037 C
0038 C      PII = 4.*ATAN(1.)
0039 C      PI8 = PII/8.
0040 C      P7 = 1./SQRT(2.)
0041 C      P7TWO = 2.*P7
0042 C      C22 = COS(PI8)
0043 C      S22 = SIN(PI8)
0044 C      PI2 = 2.*PII
0045 C      N = 1
0046 C      DO 10 I=1,15
0047 C          M = I
0048 C          N = N*2
0049 C          IF (N.EQ.NFFT) GO TO 20
0050 C      10 CONTINUE
0051 C      WRITE (IW,9999)
0052 C      9999 FORMAT(' NFFT NOT A POWER OF 2 FOR FFS')
0053 C      STOP
0054 C      20 CONTINUE
0055 C      B(2) = B(NFFT+1)
0056 C      DO 30 I=1,NFFT
0057 C          B(I) = B(I)/FLOAT(NFFT)
0058 C      30 CONTINUE

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

0059      DO 40 I=4,NFFT,2
0060          B(I) = -B(I)
0061  40      CONTINUE
0062      NSPOW = M/3
0063  C
0064  C REORDER THE INPUT FOURIER COEFFICIENTS
0065  C
0066      CALL ORD2(M, B)
0067      CALL ORD1(M, B)
0068  C
0069      IF (NSPOW.EQ.0) GO TO 60
0070  C
0071  C PERFORM THE RADIX 8 ITERATIONS
0072  C
0073      NN = N
0074      DO 50 IT=1,NSPOW
0075          INT = N/NN
0076          CALL R8SYN(INT,NN,B,B(INT+1),B(2*INT+1),B(3*INT+1),
0077      1          B(4*INT+1),B(5*INT+1),B(6*INT+1),B(7*INT+1),B(1),
0078      2          B(INT+1),B(2*INT+1),B(3*INT+1),B(4*INT+1),B(5*INT+1),
0079      3          B(6*INT+1),B(7*INT+1))
0080          NN = NN/8
0081  50      CONTINUE
0082  C
0083  C DO A RADIX 2 OR RADIX 4 ITERATION IF ONE IS REQUIRED
0084  C
0085      60      IF (M-NSPOW*3-1) 90, 80, 70
0086      70      INT = N/4
0087          CALL R4SYN(INT, B(1), B(INT+1), B(2*INT+1), B(3*INT+1))
0088          GO TO 90
0089      80      INT = N/2
0090          CALL R2TR(INT, B(1), B(INT+1))
0091      90      RETURN
0092      END

```